Display Computations

GET+MGA Entered with $T\underline{S}$ = active vehicle velocity increment

$TS_1 = \sin^{-1} (\text{unit}T\underline{S} \cdot REFSMMA\underline{T}_3)$

If $TS_1 < 0$:

$TS_1 = TS_1 + 1$ (the 1 is 360^o)

$pMGA = TS_1$

Return

GET.LVC Entered with $T\underline{S}$ = active vehicle velocity increment

$T\underline{S}_3 = - \text{unit}\underline{R}_{init}$

$T\underline{S}_2 = \text{unit}(T\underline{S}_3 * \underline{V}_{init})$

$T\underline{S}_1 = \text{unit}(T\underline{S}_2 * T\underline{S}_3)$

$$DELVLV\underline{C} = \begin{bmatrix} T\underline{S}_1 \\ T\underline{S}_2 \\ T\underline{S}_3 \end{bmatrix} \quad T\underline{S}$$

Return

CALCN83 Entered after Average-G in P47, from "TIGON" setting

$DV47TEM\underline{P} = \underline{V}_{gbody} + DELVRE\underline{F}$ ($\underline{V}_{gbody}$ notation also "DELVCTL")

$T\underline{S} = DV47TEM\underline{P}$

Perform "S41.1"

$DELVIM\underline{U} = T\underline{S}$

$VMAGI = |\underline{V}|$

$HDOT = \text{unit}\underline{R} \cdot \underline{V}$ (writes over UTYAW, R1 of N78)

$TS = K_{rpad}$

If bit 2(AMOONFLG) of FLAGWRDO = 1:

$TS = |RL\underline{S}|$, rescaled to scale factor B29

ALTI = $|\underline{R}|$ - TS (writes over UTPIT, R2 of N78, and low half of N31)

Inhibit interrupts

$\underline{V}_{gbody} = \underline{\text{DV47TEMP}}$

Proceed to "SERVEXIT" (where interrupts release as part of restart logic)

<u>CALCN85</u> Entered after Average-G in P40 (from "P40RCS" setting) and P41 (from "P40SXTY" setting)

Perform "UPDATEVG"

$\underline{TS} = \underline{V}_{gtig}$

Perform "S41.1"

$\underline{V}_{gbody} = \underline{TS}$

Proceed to "SERVXT1"

<u>SBANDANT</u> Established by "VB64" for a V64E. This is R05.

$TS_1 = T_{now}$

Perform "CDUTRIG"

$T_{decl} = TS_1$

Perform "CSMCONIC"

If X2 = 0: (means earth-centered)

$\underline{R}_{sb} = - \underline{R}_{att}$

If X2 $\neq$ 0: (means moon-centered)

$\underline{R}_{sb} = \underline{R}_{att}$

$TS = T_{att}$

Perform "LUNPOS"

$\underline{R}_{sb} = - (\underline{TS} + \underline{R}_{sb})$

$\underline{R}_{sb} = [\text{REFSMMAT}]\ \underline{R}_{sb}$

YAWANG = 0

$\underline{TS} = \underline{R}_{sb}$

Perform "*SMNB*"

$\underline{R}_{sb} = \underline{TS}$

$\underline{TS}_1 = \underline{R}_{sb} - (\underline{R}_{sb} \cdot \text{unit}\underline{Z})\ \text{unit}\underline{Z}$

$\underline{TS}_2$ = unit$\underline{TS}_1$

If $\underline{TS}_2$ does not generate overflow (e.g. a $\underline{TS}_1$ component $\geq 2^8$ meters):

YAWANG = $\cos^{-1}$ ($\underline{TS}_2$ · unit$\underline{X}$)

If ($\underline{TS}_2$ · unit$\underline{Y}$) < 0:

YAWANG = $(1 - 2^{-28})$ - YAWANG

RHOSB = $\cos^{-1}$ (unit$\underline{R}_{sb}$ · unit$\underline{Z}$) - $\frac{1}{4}$ (the $\frac{1}{4}$ is 90 degrees)

GAMMASB = YAWANG

Change priority of present job to 05_8

If bit 5 of EXTVBACT = 0: (means display response received)

Proceed to "ENDEXT"

TS = 0651_{vn}

Perform "GOMARKFR": if terminate, set bit 5 of EXTVBACT = 0 and End of job
if proceed, set bit 5 of EXTVBACT = 0 and End of job
otherwise, End of job

TS = 100_2 and perform "BLANKET" (R3BLNK)

Change priority of present job to 04_8

Delay 0.01 second (by putting job to sleep via "DELAYJOB")

Proceed to "SBANDANT"

<u>V82CALL</u> Entered from "V82PERF" for a verb 82. This is R30.

If bit 1(AVEGFLAG) of FLAGWRD1 = 1:

Establish "V82GON1" (priority 07_8, with VAC area)

Perform jobs of "higher" priority if waiting ("V82GON1" will be since VAC-area ones of a given priority are considered higher than non-VAC jobs)

TS = 1644_{vn}

Proceed to "GOXDSPF": if terminate, set bit 5 of EXTVBACT = 0 and End of job
if proceed, set bit 5 of EXTVBACT = 0 and End of job
otherwise, proceed to previous line

OPTIONX = 2 (Average-G <u>not</u> running if come here)

OPTIONX+1 = 1 (means computation for this vehicle)

TS = 0412_{vn}

Proceed to "GOXDSPF": if terminate, proceed to "ENDEXT"
if proceed, proceed
otherwise, proceed to previous line

Call "TICKTEST" in 0.08 seconds

Proceed to "V82GOFLP"

V82GOFLP

V82FLAGS = 0

Establish "V82GOFF1" (priority 07_8, with VAC area)

If bits 2-1 of V82FLAGS = 00_2: (set after data from "V82GOFF1")

Delay 1 second (by putting job to sleep via "DELAYJOB")

Proceed to second previous line (recheck bits 2-1)

TS = 1644_{vn}

Proceed to "GOXDSPF": if terminate, set bit 5 of EXTVBACT = 0
and End of job
if proceed, set bit 5 of EXTVBACT = 0
and End of job
otherwise, proceed to "V82GOFLP"

V82GON1 Established at start of "V82CALL" if Average-G on.

Set $\underline{R}_{one} = \underline{R}$ and $\underline{V}_{one} = \underline{V}$ (both from same Average-G cycle)

If bit 2(AMOONFLG) of FLAGWRDO = 0:

Set bit 13(V82EMFLG) of FLAGWRD9 = 0 (means earth-centered)

TFFdRTMU = $K_{1drtmue}$

HPERMIN = $K_{minpere}$

RPADTEM = K_{rpad}

Set bit 12(LUNLATLO) of FLAGWRD3 = 0

If bit 2(AMOONFLG) of FLAGWRDO = 1:

Set bit 13(V82EMFLG) of FLAGWRD9 = 1 (means moon-centered)

TFFdRTMU = $K_{1drtmum}$

HPERMIN = $K_{minperm}$

RPADTEM = $|RL\underline{S}|$

Set bit 12(LUNLATLO) of FLAGWRD3 = 1

Perform "SR30.1"

If MODREG = 11:

Perform "INTSTALL"

Perform "DELRSPL" (continues below if MODREG $\neq$ 0, as it should)

If bit 5 of EXTVBACT = 0: (means display response received)

Proceed to "ENDEXT"

Delay 1 second (by putting job to sleep via "DELAYJOB")

Proceed to "V82GON1"

V82GOFF1 Established by "V82GOFLP" if Average-G off.

DSPTEMX = 0

TS = 0616_{vn}

Proceed to "GOXDSPF": if terminate, proceed to "ENDEXT"
if proceed, proceed
otherwise, proceed to previous line

TS = DSPTEMX

If DSPTEMX = 0:

TS = T_{now}

T_{decl} = TS

$T_{start82}$ = TS (note that time origins remain present time, not N16 time)

If $|OPTIONX+1| \leq 1$:

Perform "CSMPREC"

If $|OPTIONX+1| > 1$:

Perform "LEMPREC"

$\underline{R}_{one} = \underline{R}_{att}$ (Tag here "BOTHSHIP")

$\underline{V}_{one} = \underline{V}_{att}$

If X2 = 0: (means earth-centered)

Set bit 13(V82EMFLG) of FLAGWRD9 = 0

TFFdRTMU = $K_{1drtmue}$

HPERMIN = $K_{minpere}$

(If X2 = 0):

RPADTEM = K_{rpad}

Set bit 12(LUNLATLO) of FLAGWRD3 = 0

If X2 $\neq$ 0: (means moon-centered)

Set bit 13(V82EMFLG) of FLAGWRD9 = 1

TFFdRTMU = $K_{ldrtmum}$

HPERMIN = $K_{minperm}$

RPADTEM = $|RL\underline{S}|$

Set bit 12(LUNLATLO) of FLAGWRD3 = 1

Perform "SR30.1"

If MODREG = 0:

Perform "INTSTALL"

Perform "DELRSPL" (to next line if MODREG = 0, as it should)

$T_{start82} = T_{now} - T_{start82}$

If mTPER = 0: (i.e. perigee not $\geqslant$ HPERMIN)

TFF = TFF + $T_{start82}$

V82FLAGS = 00002_8 (bit 2 = 1)

End of job

mTPER = mTPER + $T_{start82}$

V82FLAGS = 00001_8 (bit 1 = 1)

End of job

<u>TICKTEST</u> Called near end of "V82CALL" if Average-G not on.

If bit 5 of EXTVBACT = 0: (means display response received)

Establish "ENDEXT" (priority 25_8)

End of task

Call "TICKTEST" in 1 second

If bits 2-1 of V82FLAGS = 00_2: (set after data from "V82GOFF1")

End of task

If bits 2-1 of V82FLAGS = 01_2: (perigee $\geq$ HPERMIN)

$mTPER = mTPER + K_{1sec}$

End of task

If bits 2-1 of V82FLAGS = 10_2: (perigee not $\geq$ HPERMIN)

$TFF = TFF + K_{1sec}$

End of task (If bits = 11_2, TFF incremented by 0.03 seconds, but this should not be found)

<u>SR30.1</u> Entered from "V82GON1" and "V82GOFF1"

If bit 13(V82EMFLG) of FLAGWRD9 = 1: (means moon-centered)

$\underline{R}_{one} = \underline{R}_{one}$, shifted left 2 places (B27)

$\underline{V}_{one} = \underline{V}_{one}$, shifted left 2 places (B5)

Perform "TFFCONIC" (starting at second line)

$TS_1 = 1 + \sqrt{|1 - TFFALFA \;\; TFFNP|}$ (corresponds to $1 + e$) B2

$RPER = TFFNP \;/\; TS_1$

$TS = TS_1 \;\; TFF1dALF$

If $TS > 0$:

If $|TS| \leq K_{rapmax}$:

$RAPO = TS$

Skip next line

$RAPO = K_{rapmax}$

$TS = RAPO - RPADTEM$

If bit 13(V82EMFLG) of FLAGWRD9 = 1: (moon-centered)

TS = TS, shifted right 2 places (scale to B29)

HAPOX = TS, limited $\leq K_{maxnm}$

$TS_2 = RPER - RPADTEM$

$TS = TS_2$

If bit 13(V82EMFLG) of FLAGWRD9 = 1:

TS = TS, shifted right 2 places (scale to B29)

HPERX = TS, limited $\leq K_{maxnm}$

If TS_2 - HPERMIN $<$ 0:

mTPER = +0

If TS_2 - HPERMIN $\geq$ 0:

TS = RPER

Perform "CALCTPER"

mTPER = - TS

TS = HPERMIN + RPADTEM

Perform "CALCTFF"

TFF = - TS

Return

<u>DELRSPL</u> Entered if MODREG = 00 or 11 (from "V82GOFF1" or "V82GON1")

If TFF $\geq$ 0, proceed to "CANTDO" (have gone past interface altitude)

If TFF = -MAX, proceed to "CANTDO" (don't intersect interface alt.)

TS = $\sqrt{2\,(\text{TFRMAG} - \text{TFRTERM})/(\text{TFRMAG TFRTERM}) - \text{TFFVSQ}}$

VTERM = TS / TFFdRTMU

GAMTERM = $\cos^{-1}\left(\sqrt{\text{TFFNP}}\,/(\text{TS TFRTERM})\right)$ (|argument| limited $<$ 1)

COSDELF = 1 - 2 $\left|\text{TFFTEM} / \left(\text{TFRMAG TFRTERM } (1 + \text{TFFX})\right)\right|$

Perform "AUGEKUGL"

THETAPR = $\cos^{-1}$ (COSDELF) + PHIE

LAT = LATSPL

LONG = LNGSPL

ALT = DELVLVC_x

ALT = 0

TS = T_{pptm}

If bit 6(V37FLAG) of FLAGWRD7 = 0: (i.e. Average-G not running)

TS = $T_{start82}$

TS = TS + T_{ent} - TFF

Set bit 13(ERADCOMP) of FLAGWRD1 = 0

Perform "LALOTORV"

$RSPmRREC = THETAPR - \cos^{-1} (unit\underline{R}_{one} \cdot unitALPHA\underline{V})$ (i.e. + if overshoot)

Perform "INTWAKE" (starting at 3rd from last line, awaken jobs)

Return

CANTDO

LAT = LATSPL

LONG = LNGSPL

$ALT = DELVLVC_x$

ALT = 0

$TS = T_{pptm}$

If bit 6(V37FLAG) of FLAGWRD7 = 0:

$TS = T_{start82}$

Set bit 13(ERADCOMP) of FLAGWRD1 = 0

Perform "LALOTORV"

$RSPmRREC = - \cos^{-1} (unit\underline{R}_{one} \cdot unitALPHA\underline{V})$

Perform "INTWAKE" (starting at 3rd from last line, awaken jobs)

Return

AUGEKUGL Entered from "DELRSPL" and "RTEVN" with VTERM in OD and GAMTERM in 2D.

$TS = VTERM - K_{v21k}$

If $TS < 0$:

$PDK1 = K_{yklk2_{-10}} (VTERM - K_{v32k_{-10}}) + K_{cklk2_{-10}}$

$PDK2 = K_{yklk2_{-4}} (VTERM - K_{v32k_{-4}}) + K_{cklk2_{-4}}$

Proceed to "PHICALC-11"

$TS = TS - K_{v32k_{-6}}$ (notation also "V(3K)")

If $TS < 0$:

$PDK1 = K_{yklk2_{-8}} (VTERM - K_{v32k_{-8}}) + K_{cklk2_{-8}}$

$PDK2 = K_{yklk2_{-4}} (VTERM - K_{v32k_{-4}}) + K_{cklk2_{-4}}$

Proceed to "PHICALC-11"

$TS = TS - K_{v4k}$

If $TS < 0$:

$PDK1 = K_{yk1k2_{-8}} (VTERM - K_{v32k_{-8}}) + K_{ck1k2_{-8}}$

$PDK2 = K_{yk1k2_{-2}} (VTERM - K_{v32k_{-2}}) + K_{ck1k2_{-2}}$

Proceed to "PHICALC-11"

$TS = TS - K_{v32k_{-10}}$ (notation also "V(400)")

If $TS < 0$:

$PDK1 = K_{yk1k2_{-6}} (VTERM - K_{v32k_{-6}}) + K_{ck1k2_{-6}}$

$PDK2 = K_{yk1k2_{-2}} (VTERM - K_{v32k_{-2}}) + K_{ck1k2_{-2}}$

Proceed to "PHICALC-11"

$PDK1 = K_{yk1k2_{-6}} (VTERM - K_{v32k_{-6}}) + K_{ck1k2_{-6}}$

$PDK2 = K_{yk1k2_{0}} (VTERM - K_{v32k_{0}}) + K_{ck1k2_{0}}$

Proceed to "PHICALC-11"

PHICALC-11

$PHIE = PDK1 / (GAMTERM - PDK2)$

If overflow has not taken place:

If $PHIE \geq 0$:

If $PHIE - K_{maxphic} < 0$:

Skip next line

$PHIE = K_{maxphic}$

If $VTERM - K_{v26k} < 0$:

$T_{ent} = (K_{tless26}/VTERM)\ PHIE$

Return (to routine calling "AUGEKUGL")

$T_{ent} = K_{tgr26con}\ PHIE$

Return (to routine calling "AUGEKUGL")

R31CALL — Established by "V83PERF" and "V85PERF" for verbs 83,85: gives R31 and R34 respectively. "V83PERF" also from "P79A".

Establish "V83CALL" (priority 03_8)

Delay 1 second (by putting job to sleep via "DELAYJOB")

If bit 12 of EXTVBACT = 0: (set 1 at end of pass through "COMPDISP")

Proceed to second line of "R31CALL"

If bit 4(R31FLAG) of FLAGWRD9 = 1:

$TS = 1654_{vn}$

If bit 4(R31FLAG) of FLAGWRD9 = 0:

$TS = 1653_{vn}$

Proceed to "GOXDSPF": if terminate, set bit 5 of EXTVBACT = 0 and End of job
if proceed, set bit 5 of EXTVBACT = 0 and End of job
otherwise, proceed to 5th line of "R31CALL"

V83CALL — Established by "R31CALL"

$TS = T_{now}$ (tag here "STATEXTP")

If bit 6(V37FLAG) of FLAGWRD7 = 1:

Set $\underline{R}_{one} = \underline{R}$, $\underline{V}_{one} = \underline{V}$, and $TS = T_{pptm}$ (all from same Average-G cycle)

BASETIME = TS (Tag here "BOTHGO")

T_{decl} = BASETIME

Perform "LEMPREC"

BASEOT$\underline{P}$ = $\underline{R}_{attl}$ (B29 earth, B27 moon)

BASEOT$\underline{V}$ = $\underline{V}_{attl}$ (B7 earth, B5 moon)

NOTE: The BASEi cells are also used for VECTA$\underline{B}_i$ results in orbital integration if the MIDFLAG (bit 13 of FLAGWRD0) is set. See K_{rme_i} in Orbital Integration.

If bit 6(V37FLAG) of FLAGWRD7 = 1, proceed to "COMPDISP"

$T_{decl} = T_{att}$

Perform "CSMPREC"

BASETH$\underline{P}$ = $\underline{R}_{attl}$ (B29 earth, B27 moon)

BASETH$\underline{V}$ = $\underline{V}_{attl}$ (B7 earth, B5 moon)

Proceed to "HAVEBASE"

COMPDISP

$\underline{TS}_1 = \underline{R}_{att} - \underline{R}_{one}$

RANGE = $|\underline{TS}_1|$

Shift $\underline{TS}_1$ left by one less than number of leading zeros in RANGE

RRATE = $(\underline{V}_{att} - \underline{V}_{one}) \cdot \text{unit}\underline{TS}_1$

Perform "CDUTRIG"

TS_2+3 = CDUS

TS_2+5 = CDUT

X1 = - "TS_2"

Perform "SXTNB" (loads $T\underline{S}$)

If bit 4(R31FLAG) of FLAGWRD9 = 1:

$T\underline{S} = \text{unit}\underline{X}$

Perform "TRG*NBSM" (starting at second line)

$\underline{TS}_1 = \underline{TS}$ [REFSMMAT] (stored in 6D)

$\underline{TS}_2 = \text{unit}\left(\underline{TS}_1 - (\underline{TS}_1 \cdot \text{unit}\underline{R}_{one})\ \text{unit}\underline{R}_{one}\right)$ (stored in 12D)

$TS_3 = \left(\text{unit}(\underline{R}_{one} * \underline{V}_{one}) * \underline{R}_{one}\right) \cdot \underline{TS}_2$

RTHETA = $\cos^{-1}\left((\underline{TS}_2 \cdot \underline{TS}_1)\ \text{sgn}\ TS_3\right)$

If $(\underline{R}_{one} \cdot \underline{TS}_1) < 0$: (Note that V16 monitor could observe the uncorrected value)

RTHETA = $(1 - 2^{-28})$ - RTHETA

If bit 5 of EXTVBACT = 0: (means response received)

If MODREG $\neq$ 79:

Proceed to "ENDEXT"

Perform "CLEARMRK"

Set bit 15(PCMANFLG) of FLGWRD10 = 0

Proceed to "GOTOPOOH"

Set bit 12 of EXTVBACT = 1

Proceed to "HAVEBASE"

HAVEBASE

If bit 6(V37FLAG) of FLAGWRD7 = 1: (Average-G data available)

Set $\underline{R}_{one} = \underline{R}$, $\underline{V}_{one} = \underline{V}$, and TS = T_{pptm} (all from same Average-G cycle)

If bit 8(SURFFLAG) of FLAGWRD8 = 1:

T_{decl} = TS (tag here "GETRVN6")

Perform "LEMPREC"

Proceed to "COMPDISP"

Perform "INTSTALL"

Set bit 4(CONICINT) of FLAGWRD3 = 0

T_{decl} = TS

If bit 6(V37FLAG) of FLAGWRD7 = 0:

$T_{decl} = T_{now}$

Perform "INTSTALL"

Set bit 12(MOONFLAG) of FLAGWRD0 = 0

RC$\underline{V}$ = BASETH$\underline{P}$

VC$\underline{V}$ = BASETH$\underline{V}$

If bit 12(CMOONFLG) of FLAGWRD8 = 1:

Set bit 12(MOONFLAG) of FLAGWRD0 = 1

Set bit 4(CONICINT) of FLAGWRD3 = 0

If bit 8(SURFFLAG) of FLAGWRD8 = 0:

Set bit 4(CONICINT) of FLAGWRD3 = 1

T_{et} = BASETIME

Perform "INTEGRVS"

$\underline{R}_{one} = \underline{R}_{att}$

$\underline{V}_{one} = \underline{V}_{att}$

If bit 8(SURFFLAG) of FLAGWRD8 = 1:

$T_{decl} = T_{att}$

Perform "LEMPREC"

Proceed to "COMPDISP"

(If bit 6(V37FLAG) of FLAGWRD7 = 0):

Perform "INTSTALL"

Set bit 4(CONICINT) of FLAGWRD3 = 1

$T_{decl} = T_{att}$

Set bit 12(MOONFLAG) of FLAGWRD0 = 0 (tag here "OTHINT")

$RC\underline{V} = BASEOT\underline{P}$

$VC\underline{V} = BASEOT\underline{V}$

If bit 12(CMOONFLG) of FLAGWRD8 = 1:

Set bit 12(MOONFLAG) of FLAGWRD0 = 1

T_{et} = BASETIME

Perform "INTEGRVS"

Proceed to "COMPDISP"

R36 Established by "V90PERF" for verb 90. This is R36.

DSPTEMX = T_{ig}

TS = 0616_{vn}

Proceed to "GOXDSPF": if terminate, proceed to "ENDEXT"
if proceed, proceed
otherwise, proceed to previous line

TS = DSPTEMX

If DSPTEMX = 0:

TS = T_{now}

T_{decl} = TS

Perform "LEMCONIC" (Tag here "R36A")

$\underline{V}_{pass36} = \underline{V}_{att}$

$UN\underline{P}_{36} = unit(\underline{V}_{att} * unit\underline{R}_{att})$

$T_{decl} = T_{att}$

Perform "CSMCONIC"

$UN\underline{A}_{36} = \text{unit}(\underline{V}_{att} * \text{unit}\underline{R}_{att})$

$RANGE = \underline{R}_{att} \cdot UN\underline{P}_{36}$ (here is cross-range)

$RRATE = \underline{V}_{att} \cdot UN\underline{P}_{36}$ (here is cross-range)

$RRATE2 = \underline{V}_{pass36} \cdot UN\underline{A}_{36}$

$TS = 0696_{vn}$

Proceed to "GOXDSPF": if terminate, proceed to "ENDEXT"
if proceed, proceed to "ENDEXT"
otherwise, proceed

Proceed to second line of "R36"

S61.2 Entered from "NEWRNVN"

$TS = K_{ldrtmu}$

If $C_{emsalt} - K_{290kft} \geq 0$:

$TS = K_{ldrtmue}$

Perform "TFFCONIC"

$TS = K_{rtrial}$

Perform "CALCTFF"

$COSDELF = 1 - 2 \left| TFFTEM / \left(TFRMAG\ TFRTERM\ (1 + TFFX) \right) \right|$

$SINDELF = \left(\sqrt{1 - COSDELF^2} \right) \text{sgn}\ TFFTEM$

$UR\underline{H} = (URON\underline{E} * UN\underline{I})\ SINDELF + URON\underline{E}\ \ COSDELF$

$ALPHAV_z = UR\underline{H} \cdot \underline{C}_{unitw}$

Perform "GETERAD"

$TS = ERADM + C_{emsalt}$

Perform "CALCTFF"

$TTE1 = TS$

$TTE2 = -(TTE1 + MM)$

$TTE = TTE2 + T_{pptm}$

$COSDELF = 1 - 2 \left| TFFTEM / \left(TFRMAG\ TFRTERM\ (1 + TFFX) \right) \right|$

$SINDELF = \left(\sqrt{1 - COSDELF^2} \right) \text{sgn}\ TFFTEM$

$UR\underline{H} = (URON\underline{E} * UN\underline{I})\ SINDELF + URON\underline{E}\ \ COSDELF$

$ALPHAV_z = URH \cdot C_{unitw}$

Perform "GETERAD"

$TS = \cos^{-1} (URH \cdot R_t)$

$DTEAROT = K_{ktetal} \; TS + TTE1$

Perform "EARROT2"

$TS = \cos^{-1} (URH \cdot R_t)$

$DTEAROT = K_{ktetal} \; TS + TTE1$ (2nd time)

Perform "EARROT2"

$RTGO = \cos^{-1} (URH \cdot R_t)$

$TS = \sqrt{2(TFRMAG - TFRTERM)/(TFRMAG \; TFRTERM) - TFFVSQ}$

$VTERM = TS / TFFdRTMU$

$GAMTERM = \cos^{-1} \left(\sqrt{TFFNP}/(TS \; TFRTERM)\right)$ (argument limited < 1)

$VIO = VTERM + K_{vemscon} /(GAMTERM \; VTERM)$

$TFRTERM = ERADM + K_{minpere}$ ($K_{minpere}$ also notation "300KFT")

$TS = \sqrt{2(TFRMAG - TFRTERM)/(TFRMAG \; TFRTERM) - TFFVSQ}$

$VTERM = TS / TFFdRTMU$

$GAMAE = \cos^{-1} \left(\sqrt{TFFNP}/(TS \; TFRTERM)\right)$ (argument limited < 1)

$VBARSQ = \left((VTERM - K_{36kf}) / K_{20kf}\right)^2$

$$TS = \frac{K_{r2} \, (K_{m6p05d} + K_{r1} \; VBARSQ + GAMAE)}{(VBARSQ / K_{r4}) + 1} + K_{r3}$$

If $TS < 0$:

  $TS = 0$

$GMAX = TS$

$TS = ERADM + K_{400kft}$

Perform "CALCTFF"

$COSDELF = 1 - 2 \left| TFFTEM / \left(TFRMAG \; TFRTERM \; (1 + TFFX)\right) \right|$

$SINDELF = \left(\sqrt{1 - COSDELF^2}\right) \operatorname{sgn} TFFTEM$

$URH = (URONE * UNI) \; SINDELF + URONE \; COSDELF$

$ALPHAV_z = URH \cdot C_{unitw}$

Perform "GETERAD"

$TFRTERM = ERADM + K_{400kft}$

$TS = \sqrt{2(TFRMAG - TFRTERM)/(TFRMAG\ TFRTERM) - TFFVSQ}$

$VPRED = TS\ /\ TFFdRTMU$

$GAMMAEI = \left(-\cos^{-1}\left(\sqrt{TFFNP}\ /\ (TS\ \ TFRTERM)\right)\right)_{sp}$ (|argument| limited <1)

$JJ = RTGO_{sp}$ (for telemetry, one cell beyond GAMMAEI)

Return (to routine calling "S61.2")

TFFCONIC

$TFFdRTMU = TS$

$TFRMAG = |\underline{R}_{one}|$ (tag to enter here is "TFFCONMU")

$\underline{V}_{onepr} = TFFdRTMU\ \underline{V}_{one}$

$TFFNP = \left|(TFRMAG\ \underline{V}_{onepr}) * unit\underline{R}_{one}\right|^2$

$TFFVSQ = -\left|\underline{V}_{onepr}\right|^2$

$TFFALFA = 2\ /\ TFRMAG + TFFVSQ$

$TFFRTALF = \sqrt{|TFFALFA|}$

$TS = TFFRTALF^2\ sgn\ TFFALFA$

If $TS = 0$:

$TFF1dALF = 0$

If $TS \neq 0$:

$TFF1dALF = 1/TS$

Return

CALCTPER

Set bit 1(TFFSW) of FLAGWRD7 = 1 (means perigee time desired)

Proceed to second line of "CALCTFF"

CALCTFF

Set bit 1(TFFSW) of FLAGWRD7 = 0 (means perigee time not desired)

$TFRTERM = TS$ (TS set when enter with perigee/terminal radius)

$TS_1 = 2 - TFRTERM\ TFFALFA$

TS_2 = TFRTERM TS_1 - TFFNP

If bit 1(TFFSW) of FLAGWRD7 = 1: (means perigee time desired)

$TS_2 = 0$

If $TS_2 < 0$:

TS = +MAX (no free-fall conic to TFRTERM)

Return

TFFQ2 = $-\sqrt{TS_2}$

TFFQ1 = $\underline{V}_{onepr} \cdot \underline{R}_{one}$

If TFFQ1 < 0: (vehicle "inbound")

TFFNUM = TFRTERM - TFRMAG

TFFDEN = TFFQ1 + TFFQ2

If TFFQ1 $\geq$ 0:

TFFNUM = TFFQ2 - TFFQ1

TFFDEN = TS_1 - TFRMAG TFFALFA

If |TFFDEN| < K_{b22im}:

If TFFALFA > 0:

TFFRZ = 0

Proceed to second line of "TFFELL"

TS = 0 (near perigee for hyperbola or parabola)

Return

TS = TFFNUM TFFRTALF / TFFDEN

If |TS| $\geq$ 1:

Proceed to "TFFELL"

TFFZ = TFFNUM / TFFDEN

TFFTEM = TFFNP $TFFZ^2$ sgn TFFZ

TFFX = TFFALFA $TFFZ^2$

$$TS = K_{tx0} + K_{tx1}\ TFFX + K_{tx2}\ TFFX^2 + K_{tx3}\ TFFX^3 + K_{tx4}\ TFFX^4 + K_{tx5}\ TFFX^5$$

TS_1 = TFFZ (TFRMAG + TFRTERM - 2 $TFFZ^2$ TS)

If $TS_1 \geqslant 0$, proceed to "ENDTFF"

If TS_1 sgn TFFQ1 $\geqslant 0$, proceed to "ENDTFF"

If TFF1dALF $\leqslant 0$, proceed to "ENDTFF"

$TS_1 = TS_1 + K_{2pib5}$ TFF1dALF / TFFRTALF (corrects for period)

Proceed to "ENDTFF"

ENDTFF

If overflow has taken place since start of TFFZ calculation:

TS = +MAX

Return

TS = TFFdRTMU TS_1 (TS_1 B45 earth, B42 moon)

Return

TFFELL

TFFRZ = TFFDEN / TFFNUM

TFFX = TFF1dALF TFFRZ^2, with magnitude limited < 1.

TFFTEM = TFFNP TFF1dALF sgn (TFFRZ TFRMAG + TFFQ1)

$$TS = K_{tx0} + K_{tx1}\,\text{TFFX} + K_{tx2}\,\text{TFFX}^2 + K_{tx3}\,\text{TFFX}^3 + K_{tx4}\,\text{TFFX}^4 + K_{tx5}\,\text{TFFX}^5$$

TS_1 = 2 (TFFX TS - 1) TFFRZ TFF1dALF

$$TS_1 = \text{TFF1dALF}\left(TS_1 + \tfrac{1}{2} K_{2pib5} \sqrt{\text{TFF1dALF}} - (\text{TFFQ2} - \text{TFFQ1})\right)$$

If overflow has taken place since start of TFFTEM calculation:

TS = +MAX

Return

TS = TFFdRTMU TS_1

Return

Quantities in Computations

See also list of major variables and list of routines

ALPHA$\underline{V}$: See Coordinate Transformations. Only the Z component is required for "GETERAD".

ALT: See Coordinate Transformations. It is loaded briefly in "DELRSPL" with the X component of DELVLV$\underline{C}$ because this is the "third component" of LATSPL and LNGSPL and a vector operation is used (this transient loading might be noticed if a monitor verb were being used with a noun that displays ALT, and therefore is shown in the programmed equations).

ALTI: See Boost Computations.

BASEOT$\underline{P}$: Value of LM position vector at BASETIME computed in "V83CALL", scale factor B29(earth) or B27(moon), units meters. Subsequent computations of LM position in R31/R34 start with this position and perform precision integration (if Average-G is on) or conic integration (if Average-G is off). Cell used for VECTA$\underline{B}_6$ (page ORBI-17).

BASEOT$\underline{V}$: Value of LM velocity vector at BASETIME computed in "V83CALL", scale factor B7(earth) or B5(moon), units meters/centi-second. See BASEOT$\underline{P}$. Cell used for VECTA$\underline{B}_{18}$.

BASETH$\underline{P}$: Value of CSM position vector at BASETIME computed in "V83CALL", scale factor B29(earth) or B27(moon), units meters. If Average-G is on, the quantity is not used, but instead state vector information from Average-G is employed. If Average-G is off, however, subsequent computations of CSM position in R31/R34 start with this position and perform the integration. Cell used for VECTA$\underline{B}_{30}$.

BASETH$\underline{V}$: Value of CSM velocity vector at BASETIME computed in "V83CALL", scale factor B7(earth) or B5(moon), units meters/centi-second. See BASETH$\underline{P}$. Cell used for RPQ$\underline{V}$ (see Orbital Integration).

BASETIME: Value of time tag associated with BASEOT$\underline{P}$/$\underline{V}$ and BASETH$\underline{P}$/$\underline{V}$, scale factor B28, units centi-seconds.

C_{emsalt}: Erasable memory constant, program notation "EMSALT", scale factor B29, units meters, giving the altitude above the Fischer ellipse where the Entry Monitor System is to be initialized. The latitude argument for determining the Fischer ellipse value is determined from K_{rtrial}.

$\underline{C}_{unitw}$: See General Program Control.

CDUS, CDUT: See Optics Computations.

COSDELF: Cosine of the "transfer angle" between TFRMAG and TFRTERM, scale factor B1. Program notation "CDELF/2", stored in push-down list location 14D.

DELVIMU: Velocity increment in control axes computed in "CALCN83", scale factor B7, units meters/centi-second, used in P47 only. Value can be zeroed by entry of a recycle verb (see "P47BODY"). Cumulative sum of velocity, in reference coordinates, kept in $\underline{V}_{gbody}$ cells (note that in this case they are not control coordinates).

DELVLVC: See Burn Control.

DELVREF: See General Program Control.

DSPTEMX: Display temporary cells used to contain time for e.g. R30 and R36, scale factor B28, units centi-seconds, displayed by N16. Cells are the same as $DSPTEM2+1_{dp}$: the "X" signifies "extended verbs".

DTEAROT: See Coordinate Transformations.

DV47TEMP: Value of summed velocity increment loaded at start of "CALCN83" and transferred to $\underline{V}_{gbody}$ at end, scale factor B7, units meters/centi-second in reference coordinates. Used for restart protection of the summing process, and stored in the same cells as $\underline{V}_g$ (see Steering Computations).

ERADM: See Coordinate Transformations.

EXTVBACT: See Verb Definitions.

GAMAE: Value of flight path angle at 300,000 feet, used to compute GMAX, "positive down", stored in push-down list address 2D, units revolutions.

GAMMAEI: Single precision value of flight path angle at 400,000 feet, scale factor B0, units revolutions (will be negative). It is the "angle between the inertial velocity vector and local horizontal" at 400,000 feet above the Fischer radius. For telemetry convenience, the "least significant half" of GAMMAEI is JJ, loaded at the end of "S61.2" with $RTGO_{sp}$. See also Return to Earth Computations.

GAMMASB: Value of "yaw angle" for display in R2 of N51, scale factor B0, units revolutions, in range 0 to 360 degrees, giving S-band antenna "yaw pointing angle".

GAMTERM: Value of flight path angle e.g. at TFRTERM, scale factor B0, units revolutions, computed in a subroutine. Angle is measured as described for GAMMAEI, except that it is positive (hence could be considered as absolute value of flight path angle). When used as a communication cell with "AUGEKUGL", is retained in push-down list location 2D (and is at the same point as VTERM).

GMAX: Value of predicted maximum drag, scale factor B14, units of 0.01 g's. Computed double precision, but least significant half is not used in generating the display, so is written over by the most significant half of VMAGI after start of "CM/POSE" computations.

HAPOX: Value of apocenter altitude computed in "SR30.1", scale factor B29, units meters, displayed by N44. The "X" signifies "extended verb", since cell distinct from one in e.g. P30, HAPO.

HDOT: See Boost Computations.

HPERX: Value of pericenter altitude computed in "SR30.1", scale factor B29, units meters, displayed by N44. See HAPOX.

HPERMIN: Value of interface altitude used for R30 as the basis for computing TFF (if periapsis altitude not below this level, then mTPER is also computed). Scale factor is B29(earth) or B27 (moon), units meters.

JJ: See Entry Computations. Loaded with $RTGO_{sp}$ at end of "S61.2" for telemetry purposes (see GAMMAEI).

K_{1drtmu}: Constant, program notation "1/RTMU", scale factor B-17, value $0.5005750271E-5 \times 2^{17}$. Constant used only for initialization of TFFdRTMU in "S61.2" computations, and corresponds to the reciprocal of the square root of 3.98603223E10 x 1.0012 (with binary scale factor), where first term is earth μ ($meters^3/cs^2$) and second is a factor for "near earth trajectory", representing an increase of 0.12%. If C_{emsalt} indicates a lunar return, however, then $K_{1drtmue}$ is used.

$K_{1drtmue}$: Constant, program notation "1/RTMUE", scale factor B-17, value $0.50087529E-5 \times 2^{17}$, corresponding to reciprocal of root of unmodified earth μ (same value as K_{mutab_6} in Conic Routines).

$K_{1drtmum}$: Constant, program notation "1/RTMUM", scale factor B-14, value $0.45162595E-4 \times 2^{14}$, corresponding to reciprocal of root of moon μ (same value as $K_{mutab_{14}}$ in Conic Routines).

K_{1sec}: Constant, program notation "1SEC", scale factor B28, units centi-seconds. Value is 100×2^{-28}, corresponding to 1 second (as used). It is added to mTPER or TFF in "TICKTEST" to update time of display (if Average-G is not running) so as to reflect more closely the "time to" the indicated trajectory point (both times normally would be minus and the addition causes them to "count down").

K_{2pib5}: Constant, program notation "PI/16", scale factor B5, value $3.141592653 \times 2^{-4}$, corresponding to $2\pi \times 2^{-5}$ (also used with scale factor B4, shown as $\frac{1}{2} K_{2pib5}$, to give effect of π).

K_{20kf}: Constant, program notation "20KFT/S", scale factor B6, units meters/centi-second. Value is 121.92×2^{-7}, corresponding to $20{,}000 \times 0.3048 \times 10^{-2} \times 2^{-6}$, where first term is "basic" value in fps, second converts from feet to meters, third converts to centi-seconds, and fourth is scale factor.

K_{36kf}: Constant, program notation "36KFT/S", scale factor B7, units meters/centi-second. Value is 109.728 x 2^{-7}, corresponding to 36,000 x 0.3048 x 10^{-2} x 2^{-7}, where first term is "basic" value in fps, second converts from feet to meters, third converts to centi-seconds, and fourth is scale factor.

K_{290kft}: Constant, program notation "290KFT", scale factor B29, units meters. Value is 88392 x 2^{-29}, corresponding to 290,000 x 0.3048 x 2^{-29}, where first term is "basic" value in feet, second converts from feet to meters, and third is scale factor.

K_{400kft}: Constant, program notation "400KFT", scale factor B29, units meters. Value is 121920 x 2^{-29}, corresponding to 400,000 x 0.3048 x 2^{-29}, where first term is "basic" value in feet, second converts from feet to meters, and third is scale factor.

K_{b22im}: Constant, program notation "LIM(-22)", octal value 37777_8 37700_8, but used in program in such a way (a forcing of overflow) that effective value corresponds to 100_8, or 2^{-22}. If TFFDEN scaling is B3, this represents a "true" value of 2^{-19}; if TFFDEN scaling is B16, this represents a "true" value of 2^{-6}; and if TFFDEN scaling is B15 (moon), true value is 2^{-7}.

K_{ck1k2_i} (i = -10 to 0 in increments of 2): Value of additive constant in computing PDK1 (i = -10, -8, -6) and PDK2 (i = -4, -2, 0) in "AUGEKUGL", program notation (for i = 0) "CK1K2", scale factors all B0.
For i = 0, -2, -4, constants are in units of revolutions:

i	Value	Equivalent to
0	6.6666666E-3	2.4 degrees
-2	2.7777777E-3	1.0 degrees
-4	-8.8888888E-3	-3.2 degrees

For i = -6,-8,-10, constants are in units of revolutions/360, to give a PHIE result in units of revolutions. Conversion value between nautical miles and (rev/360) is 7.776E6, which is 21600 x 360. The 21600 in turn is 2π x 3437.7468 nmi (about 6366707 meters as contrasted with 6373338 meters for pad radius and scaling of RSPmRREC for display). Values are:

i	Value	Equivalent to
-6	3.08641975E-4	2400/7.776E6 (2400 nmi)
-8	3.08641975E-4	Same as i = -6
-10	7.07304526E-4	5500/7.776E6 (5500 nmi)

K_{ktetal}: Constant, program notation "KTETA1", scale factor B28, units centi-seconds/revolution. Value is 42.1844723 x 2^{-14}, corresponding to 1100 x 100 x 2π x 2^{-28}, where first term is "basic" value (seconds/radian), second converts to centi-seconds, third converts to revolutions, and fourth is scale factor. Cf. K_{kteta} in Entry Computations.

K_{m6p05d}: Constant, program notation "-6.05DEG", scale factor B0, units revolutions. Value is -0.016805556, corresponding to -6.05 degrees.

K_{maxnm}: Constant, program notation "MAXNM", scale factor B29, units meters. Octal value is 01065_8 05603_8, corresponding to 18519.814 kilometers, or about 9999.8995 nmi (DSKY routine display 9999.9 nmi).

$K_{maxphic}$: Constant, program notation "MAXPHIC", scale factor B0, units revolutions. Value is 0.09259298, corresponding to ~2000/21600, where first term is value in nmi and second (see K_{ck1k2}) converts to rev.

$K_{minpere}$: Constant, program notation "MINPERE", scale factor B29, units meters. Value is 91440×2^{-29}, corresponding to $300{,}000 \times 0.3048 \times 2^{-29}$, where first term in "basic" value in feet (about 49.37365 nmi), second converts to meters, and third is scale factor. Constant also referenced by tag "300KFT".

$K_{minperm}$: Constant, program notation "MINPERM", scale factor B27, units meters. Value is 10668×2^{-27}, corresponding to $35{,}000 \times 0.3048 \times 2^{-27}$, where first term is "basic" value in feet, second converts to meters, and third is scale factor.

K_{r1}: Constant, program notation "KR1", scale factor B-2, units revolutions. Value is -0.026666667, corresponding to -2.4°.

K_{r2}: Constant, program notation "KR2", scale factor B18, octal value 21450_8 00001_8. Without the non-zero least significant half, value corresponds to $4 \times 360 \times 10^2 \times 2^{-18}$ (or 9000×2^{-14}), where first term is "basic" value, second converts from revolutions to degrees, third scales to 0.01 g, and fourth is scale factor.

K_{r3}: Constant, program notation "KR3", scale factor B14, value 1000×2^{-14}. Value corresponds to $10 \times 10^2 \times 2^{-14}$, where the first term is equal to $10 - 10(x - .3)$ for x, the nominal L/D, equal to 0.3; the second term scales to 0.01 g, and the third is the scale factor: formula obtained from previous programs where $x \neq 0.3$.

K_{r4}: Constant, program notation "KR4", scale factor B-2, value 0.833333333. Since it divides VBARSQ, has an effect equivalent to multiplying it by $(5/6)^{-1} \times 2^2 = 4.80$.

K_{rapmax}: Constant, program notation "NEARONE", scale factor B29(earth) or B27(moon), units meters. Value is $(1 - 2^{-28})$, corresponding to $(2^{29} - 2)$ meters for earth and $(2^{27} - \frac{1}{2})$ meters for moon.

K_{rpad}: See Burn Control.

K_{rtrial}: Constant, program notation "RTRIAL", scale factor B29, units meters. Value stored in memory is binary equivalent of 6460098 meters: according to program comments, this is intended to be 284643 feet above a "pad" of 6373338 meters (actually is about 284645.67 feet).

$K_{tgr26con}$: Constant, program notation "TGR26CON", scale factor B28, units centi-seconds/revolution. Used to compute T_{ent} from PHIE for velocity equal to or greater than 26,000 fps (hence the notation). Value is 7.2E5 x 2^{-28}, corresponding to (1/3) x 21600 x 10^2 x 2^{-28}, where first term is "basic" value (for PHIE in nmi and answer in seconds), second converts from revolutions to nmi (see K_{ck1k2}), third converts to centi-seconds, and fourth is scale factor.

$K_{tless26}$: Constant, program notation "TLESS26", scale factor B35, value 5.70146688E7 x 2^{-35}. Used to compute T_{ent} from PHIE and VTERM for velocity less than 26,000 fps (hence the notation). Decimal value corresponds to 8660 x 21600 x 0.3048, where first term is "basic" value (for PHIE in nmi and velocity in fps), second converts from revolutions to nmi (see K_{ck1k2}), and third converts velocity distance unit from meters to feet (conversion of time unit to seconds from centi-seconds not done since time units of T_{ent} are centi-seconds).

K_{tx0} - K_{tx5}: Coefficients in a series expansion used in time-of-flight calculation, all with scale factor B0. For positive argument, function is $(\sqrt{x} - \tan^{-1}\sqrt{x})/x^{3/2}$; for negative argument, function is $(\sqrt{-x} - \tanh^{-1}\sqrt{-x})/-x^{3/2}$. According to the program comments, the polynomial was fitted for this flight over the range 0 - 1, and has a maximum error of 2E-5. Program notation "T(X) +2".

Constant	Value	Infinite-Series Value
K_{tx0}	3.333333333E-1	1/3
K_{tx1}	-1.999819135E-1	-1/5
K_{tx2}	1.418148467E-1	1/7
K_{tx3}	-1.01310997E-1	-1/9
K_{tx4}	5.609004986E-2	1/11
K_{tx5}	-1.536156925E-2	-1/13

K_{v21k}: Constant, program notation "V(21K)", scale factor B7, units meters/centi-second. Value is 64.008 x 2^{-7}, corresponding to 21000 x 0.3048 x 10^{-2} x 2^{-7}, where first term is value in fps, second converts to meters, third to centi-seconds, and fourth is scale factor.

K_{v26k}: Constant, program notation "V(26K)", scale factor B7, units meters/centi-second. Value is 79.248 x 2^{-7}, corresponding to 26000 x 0.3048 x 10^{-2} x 2^{-7}, where first term is value in fps, second converts to meters, third converts to centi-seconds, and fourth is scale factor.

K_{v32k_i} (i = -10 to 0 in increments of 2): Values of subtractive velocity constant in computing PDK1 (i = -10, -8, -6) and PDK2 (i = -4, -2,0) in "AUGEKUGL", program notation (for i = 0) "V(32K)", scale factor B7, units meters/centi-second. To convert a value in fps to meters/centi-second, multiply by $0.3048 \times 10^{-2} \times 2^{-7}$, where first term converts to meters, second converts to centi-seconds, and third is scaling.

i	Notation	Value	Equivalent to
0	V(32K)	97.536×2^{-7}	32,000 fps
-2	---	85.344×2^{-7}	28,000 fps
-4	V(24K)	73.152×2^{-7}	24,000 fps
-6	V(3K)	9.144×2^{-7}	3,000 fps
-8	V(28K)	85.344×2^{-7}	28,000 fps
-10	V(400)	1.2192×2^{-7}	400 fps

K_{v4k}: Constant, program notation "V(4K)", scale factor B7, units meters/centi-second. Value is 12.192×2^{-7}, corresponding to $4000 \times 0.3048 \times 10^{-2} \times 2^{-7}$, where first term is value in fps, second converts to meters, third converts to centi-seconds, and fourth is scale factor.

$K_{vemscon}$:Constant, program notation "VEMSCON", scale factor B14, value $-0.0389676 \times 2^{-14}$. Value corresponds to (-1.51E6) x $(0.3048/100)^2 \times (1/360) \times 2^{-14}$, where first term is "basic" value (for velocities in fps and angles in degrees), second term converts denominator to units of fps and also converts final answer from fps to meters/centi-second, third term converts angle in denominator from revolutions to degrees, and fourth term is the scale factor. Stored value is 77777_8 76601_8, corresponding to about $-0.03894043 \times 2^{-14}$: this gives a basic value of about -1.508945E6.

K_{yk1k2_i} (i = -10 to 0 in increments of 2): Value of constant multiplying velocity information in computing PDK1 (i = -10, -8, -6) and PDK2 (i = -4, -2, 0) in "AUGEKUGL", program notation (for i = 0) "YK1K2", scale factors all B-7.
For i = 0, -2, -4, basic value is for data in degrees and fps, and must be multiplied by (100/0.3048) x (1/360) to convert to proper velocity information and revolutions.

i	Value	Equivalent to
0	$2.59733157E-4 \times 2^7$	2.85E-4
-2	$9.56911636E-4 \times 2^7$	1.05E-3
-4	$1.11639691E-3 \times 2^7$	1.225E-3

For i = -6 and -10, the value of the constant is 0. For i = -8, value is $-1.86909989E-5 \times 2^7$. This value corresponds to -0.443 x (100/0.3048) x (1/7.776E6) x 2^7, where first term is "basic" value (to give a PHIE in nautical miles), second converts velocity from meters/centi-second to feet/second, third is the conversion factor discussed with K_{ck1k2}, and fourth is scale factor.

LAT: See Coordinate Transformations.

LATSPL: Input information on target latitude, program notation "LAT(SPL)", scale factor B0, units revolutions. Can be displayed in R1 of N61, and should form part of prelaunch erasable load for use in computing RSPmRREC. Cell also used to input target information for entry equations. Information is geodetic latitude.

LNGSPL: Input information on target longitude, program notation "LNG(SPL)", scale factor B0, units revolutions. Can be displayed in R2 of N61 (see LATSPL). See also Return to Earth Computations.

LONG: See Coordinate Transformations.

MM: See Entry Preparation.

mTPER: Value of time from per.alt., scale factor B28, units centi-seconds, program notation "-TPER". It is set 0 if HPERX is below HPERMIN. If V82 is entered with Average-G off, time on display is updated every second by "TICKTEST" (after being corrected near end of "V82GOFF1" for computation delay since sampling time of state vector) by being incremented by 1 second every second (hence display could become positive if perigee passed, while a termination and reselection of V82 would give a negative time again, namely time until the next perigee). If Average-G is on, however, computation is recycled every second (plus computing delays) via "V82GON1", and no updating of mTPER is otherwise performed. If HPERX is below HPERMIN, update is of TFF instead of mTPER.

OPTIONX: Cells displayed by N12, and intended for use in a fashion similar to OPTION1, OPTION2, but with extended verbs. Same cells as DSPTEMX+1. Program notation for OPTIONX+1 also "OPTIONEV".

PDK1: Value of equation quantity "K_1" (numerator term for PHIE in "AUGEKUGL" routine), units of "revolutions/360" (or revolutions^2, so that division by revolutions gives result in revolutions), scale factor B0, stored in push-down list location 4D.

PDK2: Value of equation quantity "K_2" (term in denominator for PHIE in "AUGEKUGL" routine), scale factor B0, units revolutions, stored in push-down list location 6D.

PHIE: Value of "empirically estimated entry range", scale factor B0, units revolutions, stored in push-down list location 4D. A factor of 21600 is used with constants expressed in nmi to convert to revolutions in deriving PHIE (see "AUGEKUGL" quantities).

pMGA: Value of "positive middle gimbal angle", program notation "+MGA", scale factor B0, units revolutions. Set to cause a display of -0.02 degrees if nominal flow would require a display but the bit indicating valid [REFSMMAT] is not set; set to cause a display of -0.01 degrees if computation flow does not require angle display (e.g. not the final pass through rendezvous computations). Otherwise, should be a positive number in range 0-90 degrees (or 270-360 degrees). Quantity therefore shows the value of the middle

gimbal angle (if positive) when the +X axis is aligned with the impulsive thrust direction. Cell also loaded in "PROG22" with similar information, except that angle range there is only 0-90°.

$\underline{R}_{init}$: See Rendezvous Computations. Cell also used as a communication one with "GET.LVC" to define local-vertical coordinate system.

$\underline{R}_{one}$: Position vector used to achieve a constant state vector for use within display computations. Rescaled in "SR30.1" to scale factor B27 if moon centered (otherwise, left at original B29 scaling), units meters. Also loaded in "NEWRNVN" with present position vector for use in "S61.2" computations. Also loaded in "HAVEBASE" with CSM state vector information at required time, scale factor B29, units meters, for use in "COMPDISP" for R31/R34.

$\underline{R}_{sb}$: Value of vehicle position vector towards earth computed in "SBANDANT" (initially in reference coordinates, then converted to IMU coordinates), scale factor B29, units meters, stored in push-down list location 2D (subsequently converted to navigation base coordinates). Program notations "R" and "RCM".

$\underline{R}_t$: See Coordinate Transformations.

RANGE: Value of magnitude of vector difference between LM and CSM position vectors computed in "COMPDISP" (for R31/R34), scale factor B29, units meters. Also used to contain out-of-plane position information, same units and scaling, computed in R36. Position information displayed (this cell) in R1 of N53, N54, and N96 (cell also is time-shared with other information).

RAPO: Value of apocenter radius computed in "SR30.1", scale factor B29(earth) or B27(moon), units meters, stored in push-down list location 16D.

RC$\underline{V}$: See Orbital Integration.

RHOSB: Value of "pitch angle" computed in "SBANDANT", scale factor B0, units revolutions, in range -90° to +90°. It is displayed in R1 of N51.

RL$\underline{S}$: See Coordinate Transformations.

RPADTEM: Value of base radius used in R30, scale factor B29(earth, pad radius) or B27(moon, landing site radius magnitude), units meters.

RPER: Value of pericenter radius computed in "SR30.1", scale factor B29(earth) or B27(moon), units meters, stored in push-down list location 14D.

RRATE: Value of component of velocity difference between CSM and LM vehicles (negative if approaching), scale factor B7, units meters/centi-second. Also used for CSM out-of-plane velocity information for R36, and displayed (with this tag) in R2 of N53, N54, and N96.

RRATE2: Value of LM out-of-plane velocity information computed in R36, scale factor B7, units meters/centi-second, displayed in R3 of N96. Same cell used for e.g. RTHETA.

RSPmRREC: Value of downrange recovery range error (predicted flight central angle minus target central angle, scaled to revolutions), scale factor B0, units revolutions, computed in "DELRSPL" for R30 computations in P00 or P11. Target central angle is computed as an arc cosine in the range between 0 and $\frac{1}{2}$ (0 and 180^{o}), with no account taken of flight direction. Program notation "RSP-RREC". If TFF is $\geqslant$ 0 or -MAX, the predicted flight central angle is considered 0. Cell is displayed in R1 of N50 using K_{sfotl5} (see Noun Definitions).

RTGO: Predicted central angle between the altitude specified by C_{emsalt} and splash, scale factor B0, units revolutions, computed in "S61.2". The most significant half is loaded into JJ for telemetry purposes at end of "S61.2".

RTHETA: Display angle information computed for R31 and R34, scale factor B0, units revolutions, displayed in R3 of N53 and N54. Also time-shared with other quantities (such as RRATE2).

SINDELF: Sine of the "transfer angle" between TFRMAG and TFRTERM, scale factor B1. Computed by the same subroutine as that used for COSDELF.

T_{ent}: Predicted time required to traverse the entry angle PHIE, scale factor B28, units centi-seconds, left in $MPAC_{dp}$ when exit from "AUGEKUGL" routine.

T_{et}: See Orbital Integration.

$T_{start82}$: Value of state vector time tag sampled at start of "V82GOFF1", scale factor B28, units centi-seconds. After complete "SR30.1", is replaced by (T_{now} - $T_{start82}$), or the "age" of the TFF or mTPER value which has been computed. This value is then used to update TFF or mTPER to give a time origin of present time (regardless of the time entered via N16). TFF and mTPER are then updated once a second in "TICKTEST" so as to maintain their status as time from "now" to the specified point.

TFF: Predicted time of free fall to specified altitude (negative if not yet there), scale factor B28, units centi-seconds. It is set to -MAX if trajectory does not intersect the specified altitude (either completely above or completely below it); it is set 0 for indeterminate computation (near perigee for hyperbola or parabola). Updated as discussed for mTPER (if mTPER cell is not updated, of course), for Average-G off (if on, computed on a periodic basis).

TFF1dALF: Value of |semimajor axis| computed in "TFFCONIC", program notation "TFF1/ALF", stored in push-down list location 22D. Scale factor information partially contained in X2; if X2 = 0, scale factor B22(earth) or B20(moon), units meters.

TFFALFA: Value of reciprocal of semimajor axis computed in "TFFCONIC", stored in push-down list location 26D. Scale factor information partially contained in X1; if X1 = 0, scale factor B-26 (earth) or B-24(moon).

TFFDEN: Value of denominator quantity, stored in push-down list location 0D. If TFFQ1 negative, scale factor B16 (earth) or B15(moon); if TFFQ1 positive, scale factor B3. Used as numerator in "TFFELL".

TFFdRTMU: Value of reciprocal of square root of mu of primary body, scale factor B-17(earth) or B-14(moon). If loaded in "S61.2", is reciprocal of root mu' (i.e. a mu_e multiplied by 1.0012, see K_{ldrtmu}), provided C_{emsalt} not $\geq$ 290,000 ft. Stored in 30D.

TFFNP: Value of semilatus rectum (p, called LCP, or lower-case p, in comments field) computed in "TFFCONIC", stored in push-down list location 28D. Scale factor information partially contained in X1; if X1 = 0, scale factor B38(earth) or B36(moon), units meters.

TFFNUM: Value of numerator quantity, stored in TFFX cell, scale factor B29(earth) or B27(moon) if TFFQ1 negative; scale factor B16(earth) or B15(moon) if TFFQ1 positive. Used as denominator in "TFFELL".

TFFQ1: Intermediate quantity in time-of-flight calculation (function of present state vector), scale factor B16(earth) or B15(moon), stored in push-down list location 14D.

TFFQ2: Intermediate quantity in time-of-flight calculation (function of terminal state vector), scale factor B16(earth) or B15(moon), stored in TFFTEM cell briefly.

TFFRTALF: Square root of |TFFALFA|, computed in "TFFCONIC", stored in push-down list location 24D. Scale factor information partially contained in X2; if X2 = 0, scale factor B-10(earth) or B-9(moon).

TFFRZ: Intermediate quantity ("1/Z") computed in "TFFELL", scale factor B-11(earth) or B-10(moon), stored in TFFTEM cell.

TFFTEM: Cell used for several purposes during the performance of the time-of-flight calculations; when calculation has a normal exit, contains the value of a quantity used in subsequent calculations, with scale factor information partially contained in X1. If X1 = 0, scale factor is B59(earth) or B55(moon).

TFFVSQ: Value of complement of square of velocity divided by mu, computed in "TFFCONIC", scale factor B-20(earth) or B-18(moon).

TFFX: Argument for power series expansion in time-of-flight calculation, scale factor B0, stored in push-down list location 34D.

TFFZ: Intermediate quantity ("Z") computed in case "TFFELL" is not entered, scale factor B13(earth) or B12(moon), stored in TFFTEM cell.

TFRMAG: Magnitude of $\underline{R}_{one}$ computed in "TFFCONIC" and stored in push-down list location 12D as "RMAG1", scale factor B29(earth) or B27(moon), units meters. A normalized version, with number of shifts required contained in X1, is stored in push-down list location 32D with program tag "NRMAG". Distinction in type of value employed is not made in programmed equations description.

TFRTERM: Value of terminal radius used as input argument for time-of-flight routines. With scale factor B29(earth) or B27(moon) is stored in push-down list location 18D, tag "RTERM"; shifted left by -X1 places (to give same scaling as normalized TFRMAG) is stored in NRTERM, push-down list location 16D. Distinction in type of value employed is not made in programmed equations description.

THETAPR: Value of predicted central angle of travel to target from present location, computed in "DELRSPL", scale factor B0, units revolutions. Program notation "THETA(1)".

TTE: Time from arrival at altitude specified by C_{emsalt}, scale factor B28, units centi-seconds, computed in "S61.2" and in "SERVICER" (if 05GSW bit = 0) for N63 display, negative before arrival.

TTE1: Time from arrival at altitude specified by C_{emsalt}, scale factor B28, units centi-seconds, computed in "S61.2". It is measured from the time contained in MM, and is positive before arrival.

TTE2: Complement of the GET at which vehicle will arrive at C_{emsalt}, scale factor B28, units centi-seconds, computed in "S61.2" and used in "SERVICER" to update TTE.

$UN\underline{A}_{36}$: Unit vector computed in R36 perprendicular (i.e. $\underline{v} * \underline{r}$) to the trajectory plane of the CSM (normally the "active" vehicle), scale factor B1.

$UN\underline{I}$: See Entry Computations. Computed in "NEWRNVN" also.

$UN\underline{P}_{36}$: Unit vector computed in R36 perpendicular (i.e. $\underline{v} * \underline{r}$) to the trajectory plane of the LM (normally the "passive" vehicle), scale factor B1.

$UR\underline{H}$: Value of unit terminal position vector computed in "S61.2", scale factor B2.

$URON\underline{E}$: Value of unit $\underline{R}_{one}$ computed in "NEWRNVN" for use in "S61.2", scale factor B1.

$\underline{V}_{gbody}$: See Burn Control. In "CALCN83" is used to keep cumulative sum of velocity increments in reference coordinates, with $DELVIM\underline{U}$ being the corresponding information in control coordinates. The notation also "DELVCTL".

$\underline{V}_{gtig}$: See Burn Control.

$\underline{V}_{init}$: See Rendezvous Computations. Cell also used as a communication one with "GET.LVC" to define local-vertical coordinate system.

$\underline{V}_{one}$: Velocity vector used to achieve a consistent state vector for use within display computations. Rescaled in "SR30.1" to scale factor B5 if moon centered (otherwise, left at original B7 scaling), units meters/centi-second. See $\underline{R}_{one}$.

$\underline{V}_{onepr}$: Value of $\underline{V}_{one}$ multiplied by TFFdRTMU in "TFFCONIC", scale factor B-10(earth) or B-9(moon), program notation "VONE'".

$\underline{V}_{pass36}$: Value of LM velocity vector computed in R36, scale factor B7, units meters/centi-second.

V82FLAGS: Single precision cell used to control performance of R30 (initiated by verb 82) if Average-G is off. Initialized to zero in "V82GOFLP" and set to 1 or 2 at end of "V82GOFF1" to signify that mTPER or TFF respectively should be incremented by "TICKTEST": non-zero setting also used in "V82GOFLP" to initiate V16N44 display, since such a setting signifies that data are available for display.

VBARSQ: Value of square of normalized velocity information used to compute GMAX in "S61.2", scale factor B2, stored in push-down list location OD.

VC$\underline{V}$: See Orbital Integration.

VIO: Value of predicted velocity at altitude specified by C_{emsalt}, scale factor B7, units meters/centi-second, including an approximate correction for the drag loss in reaching this altitude (the nominal 0.05 g point).

VMAGI: See Boost Computations.

VPRED: Value of predicted velocity at an altitude of 400,000 feet above Fischer radius computed in "S61.2", scale factor B7, units meters/centi-second. See also Return to Earth Computations.

VTERM: Value of velocity (from vis viva integral) at TFRTERM, scale factor B7(earth) or B5(moon), units meters/centi-second, stored in push-down list location OD (also used as communication cell with "AUGEKUGL").

YAWANG: Cell used in "SBANDANT" to contain value of yaw antenna pointing angle (0 if vectors not defined well), scale factor BO, units revolutions, loaded into GAMMASB for display. YAWANG itself is push-down list location 20D.

Digital Autopilot Entry Routines

CM/DAPON Entered from "P62"

Set bits 15-14 (DAPBIT1, DAPBIT2) of FLAGWRD6 = 00_2

T5LOC = "T5IDLOC"

T6LOC = "T5IDLOC"

Channel 6 = 0

Channel 5 = 0

Set bits 15-13 of DAPDATR1 = 0

If bit 11(GAMDIFSW) of FLAGWRD6 = 0:

Delay 0.5 second (by putting job to sleep via "DELAYJOB") and repeat check of bit 11

RCSFLAGS = 77776_8 (sets bit 3 to 1)

P63FLAG = -1

JETAG = 0

PAXERR1 = 0

Inhibit interrupts

ALFACOM = ALFAd180

BETACOM = BETAd180 ("least significant half" of ALFAi)

ROLLHOLD = ROLLd180

$ROLLC_{sp}$ = ROLLd180 (rescaled to B0 revolutions)

Set bit 12(CMDAPARM) of FLAGWRD6 = 1

Set bit 1(NODOV37) of FLAGWRD2 = 1

Release interrupts

Proceed to "P62.1"

READGYMB Called initially by "P62"

CMdGYMDT = CMdGYMDT + 10

If bit 6(NOIMUDAP) of IMODES33 = 1: (IMU data not usable)

Set bit 1(GYMDIF) of FLAGWRD6 = 0

(If bit 6 of IMODES33 = 1):

Channel 6 = 0

Channel 5 = 0

JETAG = 0

OLDELi = 0 (i = P,Q,R)

GAMDOT = 0

Call "READGYMB" in 0.1 second

End of task

Set AOG = CDU_x and TS = AOG

mDELAOG = TS - AOG (ones complement difference formed)

Set AIG = CDU_y and TS = AIG

mDELAIG = TS - AIG (ones complement difference formed)

Set AMG = CDU_z and TS = AMG

mDELAMG = TS - AMG (ones complement difference formed)

If bit 2(CMDSTBY) of FLAGWRD6 = 0:

Channel 6 = 0

Channel 5 = 0

End of task

If bit 1(GYMDIF) of FLAGWRD6 = 0:

Set bit 1(GYMDIF) of FLAGWRD6 = 1

JETAG = 0

OLDELi = 0 (i = P,Q,R)

GAMDOT = 0

Call "READGYMB" in 0.1 second

End of task

Call "READGYMB" in 0.1 second

Proceed to "BODYRATE"

BODYRATE

$TS = - mDELAMG \sin_{sp} AOG - mDELAIG \cos_{sp} AOG \cos_{sp} AMG$

$TS_1 = TS - OLDELQ$

$OLDELQ = TS$

$QREL = TS + \frac{1}{2} TS_1$

$TS = mDELAIG \sin_{sp} AOG \cos_{sp} AMG - mDELAMG \cos_{sp} AOG$

$TS_1 = TS - OLDELR$

$OLDELR = TS$

$RREL = TS + \frac{1}{2} TS_1$

$TS = - mDELAOG - mDELAIG \sin_{sp} AMG$

$TS_1 = TS - OLDELP$

$OLDELP = TS$

$PREL = TS + \frac{1}{2} TS_1$

If GAMDOT $\neq$ 0:

$PREL = PREL + K_{sintr} GAMDOT \sin_{sp} (- ROLLd180)$

$QREL = QREL - GAMDOT \cos_{sp} ROLLd180$

$RREL = RREL - K_{costr} GAMDOT \sin_{sp} (- ROLLd180)$

If bit 12(CMDAPARM) of FLAGWRD6 = 0, End of task

Set TIME5 to cause program interrupt #2 in 10 milliseconds

T5LOC = "ATTRATES"

End of task

ATTRATES

ALFAd180 = ALFAd180 + QREL, in range $\pm$ 180°

CALFA = $\cos_{sp}$ ALFAd180

SALFA = $\sin_{sp}$ ALFAd180

PHIDOT = PREL CALFA + RREL SALFA

BETADOT = RREL CALFA - PREL SALFA

BETAd180 = BETAd180 + BETADOT

ALFAd180 = ALFAd180 + PHIDOT $\sin_{sp}$ BETAd180, in range $\pm$ 180°

AK_1 = ALFACOM - ALFAd180, in range $\pm$ 180°

QAXERR = AK_1

ROLLTM = ROLLd180 + PHIDOT, in range $\pm$ 180°

ROLLd180 = ROLLTM

RAXERR = BETACOM - BETAd180

If bit 3(05GSW) of FLAGWRD6 = 0, proceed to "EXDAP"

CMDAPMOD = -1

$AK_1 = 77776_8$ (one bit negative)

$AK_2 = 77776_8$ (one bit negative)

TS_1 = RREL - K_{sintr} PREL

TS = 1 sgn TS_1

If $|TS_1| - K_{ydtim} < 0$:

TS = 0 (inside deadband)

$TS_3 = K_{yjetcd_{TS}}$

TS = 1 sgn QREL

If $|QREL| - K_{ydtim} < 0$:

TS = 0 (inside deadband for pitch rate damping)

Proceed to "EXDAPIN"

EXDAPIN

Channel 5 = $TS_3 + K_{prjcd_{TS}}$

If JETAG $\nless$ 0:

If JETAG = 0:

PAXERR1 = PAXERR1 - CALFA PHIDOT

Proceed to "CM/FDAIR"

JNDX = - 1

mVTd180 = - K_{2tdtcd} PREL (scaled B-2 revolutions/DAP cycle)

TS = - ROLLd180, rescaled to scale factor B0 revolutions

If bit 4(LATSW) of FLAGWRD6 = 1:

$TS_1 = \frac{1}{2} + \frac{1}{2} + ROLLC_{sp}$ modulo 1 (the 1 is 360°)

$TS_2 = TS - \frac{1}{2} - \frac{1}{2}$

LCXd360 = $TS_1 + TS_2$ modulo 1

TS = $K_{180d8att}$ mVTd180 |mVTd180| / CALFA + LCXd360 + $\frac{1}{2}$ sgn LCXd360
(sgn sets term 0 if LCXd360 = 0)

If |TS| $\geq$ 1: (the 1 is 360°)

LCXd360 = LCXd360 - 1 sgn TS

If bit 4(LATSW) of FLAGWRD6 = 0:

Set bit 4(LATSW) of FLAGWRD6 = 1

LCXd360 = $ROLLC_{sp}$ + TS modulo 1

LCXd360 = LCXd360 CALFA

If CMDAPMOD = 1:

JETAG = 0

Proceed to "CM/FDAIR"

If CMDAPMOD = -0:

LCXd360 = 0

$ERROR_2$ = LCXd360 (for telemetry)

PAXERR1 = LCXd360

Rescale mVTd180 to scale factor B-1 revolutions/DAP cycle.

mVTd180E = mVTd180

$VSQd4API = K_{180d8att}\ mVTd180^2$

$TS = LCXd360 + K_{xsd360} - (K_{xsd360} - VSQd4API)\ \mathrm{sgn}\ mVTd180$

(if mVTd180 = 0, sgn term set 0)

If $TS \leqslant 0$: (point in left half plane)

mVTd180 = - mVTd180

LCXd360 = - LCXd360

JNDX = - JNDX

Proceed to "DZ1"

If $2\ K_{xsd360} - TS \leqslant 0$, proceed to "DZ1" (point in right half plane)

If $VSQd4API - K_{vsqmn} \leqslant 0$:

If bit 1 of CMDAPMOD = 1: (i.e. -0, since +1 already exit)

ROLLHOLD = ROLLTM

If bit 1 of CMDAPMOD = 0: (i.e. +0 or -1)

$ROLLHOLD = ROLLC_{sp}$, rescaled to B-1 revolutions in range $\pm\ 180^{\circ}$

Channel 6 = 0

VDTd180 = 0 (point in velocity deadzone)

JETAG = 0

Proceed to "CM/FDAIR"

JNDX1 = - JNDX (point in buffer zone)

VDTd180 = - mVTd180

$T_{on1} = 0$

Proceed to "GETON2"

<u>DZ1</u>

If $VSQd4API - K_{vsqmn} \leqslant 0$:

If $LCXd360 + mVTd180 - K_{xmind360} \leqslant 0$:

If bit 1 of CMDAPMOD = 1:

ROLLHOLD = ROLLTM

If bit 1 of CMDAPMOD = 0:

$ROLLHOLD = ROLLC_{sp}$, rescaled to B-1 revolutions in range $\pm\ 180^{\circ}$

(If $LCXd360 + mVTd180 - K_{xmind360} \leq 0$):

Channel 6 = 0

VDTd180 = 0

JETAG = 0

Proceed to "CM/FDAIR"

JNDX1 = - JNDX

$XDd360 = \frac{1}{2}\ VSQd4API + LCXd360 - K_{xsd360}$

$TS_2 = XDd360 + K_{mvmd360k}$

If $TS_2 < 0$:

$VDTd180 = K_{ktrcs}\ XDd360$

QREG = ZREG (positive non-zero)

If $TS_2 \geq 0$:

QREG = +0

$VDTd180 = - K_{mvmd360k}$ (notation also "-VMT/180")

$T_{on1} = K_{180d8att}\ (VDTd180 + mVTd180)$

If $T_{on1} \leq 0$:

If QREG > 0:

VDTd180 = - mVTd180

$T_{on1} = 0$

If QREG = 0:

JNDX = JNDX1

$T_{on1} = - T_{on1}$

Proceed to "GETON2"

GETON2

$T_{on2} = 2\ K_{180d8att}\ VDTd180$

If $T_{on2} < 0$:

$T_{on2} = - T_{on2}$

JNDX1 = JNDX

If VDTd180 = 0:

$T_{off} = 2\ K_{2jett}$

If $VDTd180 \neq 0$:

$$TS = LCXd360 + \frac{1}{2} T_{on1} (mVTd180 - VDTd180) - \frac{1}{2} T_{on2} VDTd180$$

If $|TS| - 2 |VDTd180| \geqslant 0$: (implemented by division of TS shifted right 14 places by VDTd180 and check of A: the "2" comes about from TS scaling of B0 and VDTd180 of B-1)

$$T_{off} = 2 K_{2jett}$$

If $|TS| - 2 |VDTd180| < 0$:

$$T_{off} = K_{2jett} \quad TS / VDTd180$$

$T_{on1} = K_{4jett} T_{on1}$

$T_{on2} = K_{4jett} T_{on2}$

JETAG = 0

SWdNDX = 1

TUSED = TUSED + TIME1 + 8192 + 8192, modulo 2^{14}

TUSED = TUSED + K_{mtm3}

If $T_{on1} - 2 \leqslant 0$: (The "2" in this and subsequent equations is in units of centi-seconds, as are the times).

$T_{on1} = -1$

If $T_{on1} - 2 > 0$:

T1BITS = $K_{prjcd_{JNDX}}$

TUSED = TUSED + T_{on1}

If TUSED > 0:

$T_{on1} = 0$

$T_{off} = -1$

$T_{on2} = -1$

Proceed to "JETCALL1"

If $T_{off} - 2 \leqslant 0$:

$T_{off} = -1$

If $T_{off} - 2 > 0$:

$TUSED = TUSED + T_{off}$

If $TUSED > 0$:

$T_{off} = 0$

$T_{on2} = -1$

Proceed to "JETCALL1"

If $T_{on2} - 2 \leq 0$:

$T_{on2} = -1$

Proceed to "JETCALL1"

$T2BITS = K_{prjcd_{JNDX1}}$

$TUSED = TUSED + T_{on2}$

If $TUSED > 0$:

$T_{on2} = 0$

Proceed to "JETCALL1"

JETCALL1

OUTTAG = 0

NUJET = 0

TBITS = 0

Set $TS_1 = T_{on1}$ and $T_{on1} = 0$

$TS_2 = T1BITS$ (T1BITS written over with meaningless data)

If $TS_1 > 0$, proceed to "JETCALL2"

If $TS_1 = 0$, proceed to "JETCALL3"

If $TS_1 < 0$, proceed to second line of "JETCALL3"

JETCALL3

$NUJET = TS_2$

Set $TS_1 = T_{off}$ and $T_{off} = -1$

$TS_2 = TBITS$ (TBITS written over with meaningless data)

If $TS_1 > 0$, proceed to "JETCALL2"

If $TS_1 = 0$:

NUJET = TS_2

Set $TS_1 = T_{on2}$ and $T_{on2} = -1$

TS_2 = T2BITS (T2BITS written over with meaningless data)

If $TS_1 > 0$, proceed to "JETCALL2"

If $TS_1 = 0$:

NUJET = TS_2

Proceed to third line of "JETCALL2"

<u>JETCALL2</u>

NUJET = NUJET + TS_2

Call "JETCALL" in TS_1 centi-seconds

Channel 6 = NUJET

If OUTTAG > 0, End of task

Proceed to "CM/FDAIR"

<u>JETCALL</u>

OUTTAG = 2

If bit 2(CMDSTBY) of FLAGWRD6 = 1:

$TS_2 = 0$

Proceed to "JETCALL3"

Channel 6 = 0

End of task

<u>CM/FDAIR</u>

$AK_0 = \frac{1}{4}$ PAXERR1 (gives full scale of $67\frac{1}{2}°$)

If SWdNDX < 0:

SWdNDX = - SWdNDX

Perform "NEEDLER"

T5LOC = "T5IDLOC"

Resume

SWdNDX = 3 - SWdNDX

If SWdNDX > 0:

CMTMTIME = TIME1

SWdNDX = - 13

TS = SWdNDX - 1

$ENDBUF_{TS}$ = (PREL, QREL, RREL)

T5LOC = "T5IDLOC"

Resume

EXDAP

CMDAPMOD = +0

If |CALFA| - K_{c45im} < 0: (i.e. in range 45 to 135 degrees)

CMDAPMOD = +1

JETAG = -1

If P63FLAG = 1, set P63FLAG = 0

TS_1 = - RAXERR sgn SALFA (0 for SALFA = 0)

PAXERR1 = TS_1 , rescaled to scale factor B0, units revolutions

TS_2 = - BETADOT sgn SALFA (0 for SALFA = 0)

Perform "BIASEDZ"

Channel 6 = $K_{prjcd_{TS}}$

RAXERR = (ROLLHOLD - ROLLd180) sgn SALFA (0 for SALFA = 0)

If |CALFA| - K_{c45im} ≥ 0:

If CALFA > 0:

If P63FLAG = 0:

P63FLAG = -1

Call "WAKEP62" in K_{nsec} centi-seconds

If JETAG ≤ 0:

JETAG = 0

Channel 6 = 0

If CALFA < 0:

CMDAPMOD = -0

RAXERR = RAXERR sgn CALFA

AK_2 = RAXERR

TS_1 = RAXERR

TS_2 = RREL

If CMDAPMOD = +0:

TS_2 = BETADOT

Perform "BIASEDZ"

$TS_3 = K_{yjetcd_{TS}}$

TS_1 = QAXERR

TS_2 = QREL

Perform "BIASEDZ"

Proceed to "EXDAPIN"

BIASEDZ Entered with attitude information in TS_1 (L-register) and rate information in TS_2 (A-register).

$TS_1 = TS_1 - K_{cmdbs} \text{ sgn } TS_2$ (gives biased error; sgn sets term zero if $TS_2 = 0$)

$TS = -1 \text{ sgn } TS_1$

If $|TS_1| - K_{yawim} < 0$:

$TS = 0$

If $|TS_2| - K_{4dsim} \geq 0$:

$TS = 0$

$TS_4 = 1 \text{ sgn } TS_2$

If $|TS_2| - K_{ydtim} < 0$:

$TS_4 = 0$

$TS = TS + TS_4$, with magnitude limited ≤ 1 (done in calling routine by an "or" function)

Return

Quantities in Computations

See also list of major variables and list of routines

AIG: Value of CDU_y sampled by "READGYMB", single precision twos complement, scale factor B-1, units revolutions ("inner gimbal angle").

AK_i (i = 0-2): See Digital Autopilot Interface Routines.

ALFACOM: Single precision value of commanded "pitch" angle, scale factor B-1, units revolutions. Set to ALFAd180 in "CM/DAPON", and to $C_{alfapad}$ in "P62.1" (see Entry Preparation).

ALFAd180: See Entry Computations.

AMG: Value of CDU_z sampled by "READGYMB", single precision twos complement, scale factor B-1, units revolutions ("middle gimbal angle").

AOG: Value of CDU_x sampled by "READGYMB", single precision twos complement, scale factor B-1, units revolutions ("outer gimbal angle").

BETACOM: Single precision value of commanded "yaw" angle, scale factor B-1, units revolutions. Set to BETAd180 in "CM/DAPON", and to zero in "P62.1".

BETAd180: See Entry Computations.

BETADOT: Single precision value of rate of change of "yaw" angle, scale factor B-1, units revolutions/deci-second.

CALFA: Single precision value of cos ALFAd180, scale factor B0, after ALFAd180 updated only by QREL (see "ATTRATES").

CMDAPMOD: Single precision quantity used to control the performance of the entry DAP. Individual values have the following meanings:

Value	Significance
-1	Bit 3(05GSW) of FLAGWRD6 = 1 (value set in "ATTRATES").
-0	Bit 3(05GSW) of FLAGWRD6 = 0 and CALFA is at least K_{c45im} in magnitude with negative sign (causes damping only, with LCXd360 set to 0). Value set in "EXDAP".
+1	Bit 3(05GSW) of FLAGWRD6 = 0 and CALFA magnitude less than K_{c45im}. Value set in "EXDAP".
+0	Bit 3(05GSW) of FLAGWRD6 = 0 and CALFA at least K_{c45im} in magnitude with positive sign. Value set in "EXDAP".

Scale factor is B14.

CMdGYMDT: Single precision call time, scale factor B14, units centi-seconds, used to permit proper restart time to be set for performance of "READGYMB". Initialized to 5 in "READACCS" (when the base time used for the restart is also set), giving "READGYMB" performance 5 cs out of phase with "READACCS".

CMTMTIME: Value of TIME1 loaded for telemetry purposes in "CM/FDAIR", scale factor B14, units centi-seconds. Cell is the same as UPBUFF+0 (see Uplink Processing).

DAPDATR1: See Digital Autopilot Interface Routines.

ENDBUF: Set of cells used for telemetry of (PREL, QREL, RREL), loaded in "CM/FDAIR" on alternate entries to entry DAP (every 0.2 seconds the cell set is loaded). Cell set loaded is indexed by a negative quantity, with ENDBUF-14, ENDBUF-11, etc. used for successive PREL values. ENDBUF-14 is the same cell as UPBUFF+2 (see Uplink Processing).

$ERROR_2$: See Digital Autopilot RCS Routines (loaded in "EXDAPIN" with the original value of LCXd360 (before sign reversal, if any), for telemetry).

GAMDOT: See Entry Computations.

IMODES33: See IMU Computations.

JETAG: Single precision quantity, scale factor B14, used to control the performance of the entry DAP. The "normal" value is zero; a setting to +1 is made by "SETJTAG" (called every two seconds by "READACCS") to cause an update of roll output to be made; a setting to -1 is made in "EXDAP" if the magnitude of CALFA is less than K_{c45im}.

JNDX: Value of jet table index for time interval specified by T_{on1}, scale factor B14. Initialized to -1 near start of "EXDAPIN", and complemented subsequently if necessary. A negative value causes positive jets to be selected by indexing K_{prjcd}. The quantity is single precision.

JNDX1: Single precision quantity, scale factor B14, serving a similar function to JNDX, but for the jets controlled by the time interval specified by T_{on2}.

K_{2jett}: Single precision constant, program notation "2JETT", scale factor B13, units centi-seconds/entry DAP cycle. Value is 400×2^{-14}, corresponding to 200×2^{-13}, i.e. a two-second entry DAP cycle (for performance of roll command updates).

K_{2tdtcd}: Single precision constant, program notation "2T/TCDU", scale factor B13, units (entry DAP period)/(CDU sample period). Value is 40×2^{-14}, corresponding to 20×2^{-13}, i.e. a two-second entry DAP cycle (to perform roll command updates) and a 0.1 second CDU sampling period (period of performance of "READGYMB"). Converts angular rates from units of revolutions/deci-second to revolutions/DAP cycle.

K_{4dsim}: Single precision constant, program notation "4D/SLIM", scale factor B-1, units revolutions/deci-second. Value is 16348 x 2^{-14}, but used in the program in such a way (a check for overflow, i.e. magnitude of 16384 or more, on an argument decremented by 1) that the effective value is (16384 - 16348 + 1), or 37 least increments, corresponding to 37 x 180° x 2^{-14} x 10 = 4.065°/sec. The argument is decremented by 1 for convenience in forming the absolute value.

K_{4jett}: Single precision constant, program notation "4JETT", scale factor B12, units centi-seconds/entry DAP cycle. Value is 800 x 2^{-14}, corresponding to 200 x 2^{-12} (cf. K_{2jett}).

$K_{180d8att}$: Single precision constant, program notation "180/8ATT", scale factor B4, units of (revolutions/DAP cycle^2)$^{-1}$. Nominal value is 0.61813187, corresponding to the reciprocal of: (9.1 x (1/360) x 2^2 x 2^4), where first term is acceleration (degrees/second^2), second converts to units of revolutions, third converts to units of DAP cycle^2 (2 seconds each), and fourth is scale factor. Actual stored value is about 0.61810. Alternatively, 9.1 = 2 x 4.55.

K_{c45im}: Single precision constant, program notation "C45LIM", scale factor B0. Value is 11277_8 = 4799, but used in program in such a way (a check for overflow, i.e. a magnitude of 16384 or more, on an argument decremented by 1 for convenience in forming the absolute value) that effective value is (16384 - 4799 + 1) = 11586 x 2^{-14} = 0.707153, or about cos 45°.

K_{cmdbs}: Single precision constant, program notation "CM/BIAS", scale factor B-1, units revolutions. Value is 55 x 2^{-14}, corresponding to about 0.604°.

K_{costr}: Single precision constant, program notation "COSTRIM", scale factor B0, value 0.93969, corresponding to cos (-20°).

K_{ktrcs}: Single precision constant, program notation "KTRCS", scale factor B0, value 0.5. Value corresponds to 0.25 x 2, where first term is "K"("slope of line determining rate at which error reduced") and second is DAP roll cycle rate of 2 seconds.

K_{mtm3}: Single precision constant, program notation "-T-3", scale factor B14, units centi-seconds. Value is -203 x 2^{-14}, corresponding to -2.03 seconds. See TUSED.

$K_{mvmd360k}$: Single precision constant, program notation "-VM/360K", scale factor B0. Nominal value is -0.22222222, corresponding to (-1) x 20 x (1/360) x (1/0.25), where first term is an equation factor, second is "maximum roll rate limit" (degrees/second), third converts to revolutions, and fourth is reciprocal of "K" (cf. K_{ktrcs}). Program notation also "-VMT/180", for which would have scale factor B-1, units revolutions/DAP cycle, where the "(1/0.25)" factor is 2 x 2^1 (the first factor is two-second cycle, the 2nd is scale).

K_{nsec}: Single precision constant, program notation "NSEC", scale factor B14, units centi-seconds. Value is 2100 x 2^{-14}, corresponding to 21 seconds. Program comments indicate it is an approximation to the time required to travel 65° at a rate of 3°/second.

K_{prjcd}: Set of three single precision cells, program notation "P/RJCODE", selected by means of an index having values -1, 0, +1. Values are:

Index	Cell	Channel 6 Jets	Channel 5 Jets
-1	00005_8	#11, #9 (+ R)	#3, #1 (+ P)
0	00000_8	none	none
+1	00012_8	#10, #12(- R)	#2, #4 (- P)

K_{sintr}: Single precision constant, program notation "SINTRIM", scale factor B0, value -0.34202, corresponding to sin (-20°). Program comments indicate that this is trim angle for L/D = 0.3.

K_{vsqmn}: Single precision constant, program notation "VSQMIN", scale factor B1, units revolutions. Nominal value is 0.61050061E-3, corresponding to $(2/360)^2 \times (9.1/360)^{-1} \times 2^{-2} \times 2^2 \times 2^{-1}$, where first term is value of the square of "VMIN" (expressed in units of revolutions), second is acceleration (in revolutions/ second^2), third converts acceleration to DAP cycles (cf. $K_{180d8att}$), fourth converts VMIN to units of DAP cycles (from rev/sec), and fifth is scale factor. Actual value is about 0.61035E-3, i.e. 00012_8.

$K_{xmind360}$: Single precision constant, program notation "XMIN/360", scale factor B0, units revolutions. Value is 182×2^{-14}, corresponding approximately to $4 \times (1/360)$, where first term is value of "XMIN" (degrees), and second converts to revolutions.

K_{xsd360}: Single precision constant, program notation "XS/360", scale factor B0, units revolutions. Value is 91×2^{-14}, corresponding approximately to $2 \times (1/360)$, where first term is value of "XS" (degrees), and second converts to revolutions. Tag also "BUFLIM".

K_{yawim}: Single precision constant, program notation "YAWLIM", scale factor B-1, units revolutions. Value is 16055×2^{-14}, but used in the program in such a way (a check for overflow, i.e. a magnitude of 16384 or more, on an argument decremented by 1 for convenience in forming the absolute value) that effective value is (16384 - 16055 + 1) = 330 least increments, corresponding to $330 \times (360) \times 2^{-15} = 3.625^{\circ}$.

K_{ydtim}: Single precision constant, program notation "YDOTLIM", scale factor B-1, units revolutions/deci-second. Value is 16366×2^{-14}, but used in program in such a way (cf. K_{4dsim}) that effective value is 19 least increments, corresponding to about 2.087°/second.

K_{yjetcd}: Set of three single precision cells, program notation "YJETCODE", selected by means of an index having values -1, 0, +1. Values are:

Index	Cell	Channel 5 Jets
-1	00120_8	#7, #5 (+ Y)
0	00000_8	none
+1	00240_8	#6, #8 (- Y)

LCXd360: Single precision value of roll attitude error (shortest path if LATSW = 1 when enter computations, otherwise an enforced roll over the top) to be corrected, scale factor B0, units revolutions.

mDELAIG: Complement of change in AIG during previous 0.1 second, scale factor B-1, units revolutions/deci-second: is $AIG_{n-1} - AIG_n$.

mDELAMG: Complement of change in AMG during previous 0.1 second, scale factor B-1, units revolutions/deci-second. Notation "-DELAMG".

mDELAOG: Complement of change in AOG during previous 0.1 second, scale factor B-1, units revolutions/deci-second.

mVTd180: Complement of value of PREL, converted to units of revolutions/ DAP entry cycle (of 2 seconds). Initially computed with scale factor B-2, and subsequently rescaled to scale factor B-1. Program notation is "-VT/180". Complemented if point in left half plane.

mVTd180E: Value of mVTd180 for telemetry, scale factor B-1, original sign.

NUJET: Single precision cell used to contain the jet pattern to be loaded into channel 6 in "JETCALL2".

OLDELi (i = P,Q,R): Previous values of "raw" PREL, QREL, and RREL, scale factor B-1, units revolutions/deci-second, used in "BODYRATE" to compensate for angular accelerations. The values used are the uncompensated PREL, QREL, and RREL (i.e. those derived based solely on CDU measurement data this cycle and the previous cycle).

OUTTAG: Single precision quantity, scale factor B14, set to 2 if "JETCALL" performed and to 0 if "JETCALL1" performed, indicating respectively that the exit from "JETCALL2" should be to End of task and to "CM/FDAIR".

P63FLAG: Single precision cell, scale factor B14, used for control of entry DAP sequencing. It is set to -1 in "CM/DAPON" and in "EXDAP" after "WAKEP62" calling computations initiated. It is set to 1 in "P62.1" after a response to the V06N61 display, and if 1 is set 0 in "EXDAP" if |CALFA| is less than K_{c45im} (a 0 value permits "WAKEP62" calling computations). Intended to prevent P63 from starting until response to V06N61 display, and to avoid more than one "WAKEP62" call. It is set to -1 in "P63".

PAXERR1: Single precision quantity, scale factor B0, units revolutions, containing the information to be loaded into AK_0: during atmospheric entry, it is set to LCXd360 each two seconds, and updated by (CALFA) x (- PHIDOT) each 0.1 second between two-second roll updates.

PHIDOT: Single precision value of rate of change of "roll" angle, scale factor B-1, units revolutions/deci-second. Cos Beta $\cong$ 1.

PREL: Single precision value of "roll" body angular velocity, scale factor B-1, units revolutions/deci-second. It includes corrections for acceleration and GAMDOT, and is along the "$\underline{X}_b$ body axis".

QAXERR: Single precision value of pitch attitude error (saved for telemetry purposes), scale factor B-1, units revolutions. It is also used in "EXDAP", and is computed in "ATTRATES".

QREG: Computer single precision Q register (address 0002_8), used to retain program control information in "DZ1" (zero or non-zero value). After a TC (transfer control) order, Q is loaded with the S-register contents corresponding to the address of the following step (see 3420.5-27).

QREL: Single precision value of "pitch" body angular velocity, scale factor B-1, units revolutions/deci-second (including corrections for acceleration and GAMDOT), measured along the "$\underline{Y}_b$ body axis".

RAXERR: Single precision value of yaw attitude error (saved for telemetry purposes), scale factor B-1, units revolutions. It is loaded in "EXDAP" and "ATTRATES".

RCSFLAGS: See Digital Autopilot Interface Routines.

ROLLC: See Entry Computations.

ROLLd180: See Entry Computations.

ROLLHOLD: Single precision value of ROLLd180 sampled when the entry DAP is started (in "CM/DAPON"), scale factor B-1, units revolutions, and used to provide error signal information for RAXERR in "EXDAP". It is set to 0 in "KEP2", and to ROLLC or ROLLTM in "EXDAPIN"/"DZ1".

ROLLTM: Value of ROLLd180 placed in a special cell for telemetry purposes, scale factor B-1, units revolutions, single precision.

RREL: Single precision value of "yaw" body angular velocity, scale factor B-1, units revolutions/deci-second (including corrections for acceleration and GAMDOT), measured along the "$\underline{Z}_b$ body axis".

SALFA: Single precision value of sin ALFAd180, scale factor B0, after ALFAd180 updated only by QREL (see "ATTRATES").

SWdNDX: Single precision cell, scale factor B14, used for indexing the proper ENDBU$\underline{F}$ cells to be loaded and to cause alternation in "CM/FDAIR" between update of attitude error needles and loading of ENDBU$\underline{F}$. It is set to 1 in "P62" and also in "GETON2", causing ENDBU$\underline{F}$ to be loaded (after initialization to -13). Negative values cause "NEEDLER" to be performed (by "CM/FDAIR"), and the cell is complemented each entrance to "CM/FDAIR". The cell is the same as UPBUFF+1 (see Uplink Processing).

T_{off}: Single precision value of time, scale factor B14, units centi-seconds, required for "drifting" (between firing times of jets specified by T_{on1} and T_{on2}). Corresponding "jet pattern" (zero) is in TBITS. Cell set 0 if no waitlist call is to be made (i.e. if T_{on1} is small but the T_{off} value is such that the time delay would expire after the next two-second computing interval); cell set -1 if the time interval specified should be ignored (i.e. is too small or has already been used in "JETCALL3" or T_{on1} large).

T_{on1}: Single precision value of time, computed initially with scale factor B2, units DAP roll cycles, and subsequently rescaled to scale factor B14, units centi-seconds (by using K_{4jett}), giving the required "on time" for the jets in T1BITS. Cell set 0 if no waitlist call to be made (T_{on1} of such a size that jets should be left on for the full two-second DAP roll computing interval); cell set to -1 if the time interval involved should be ignored (i.e. is too small).

T_{on2}: Single precision value of time, same scaling as T_{on1}, giving the required "on time" for the jets in T2BITS. Cell set 0 if no waitlist call is to be made(because delay would expire after the next two-second computing interval); set to -1 if the time involved should be ignored (i.e. is too small, if $T_{on1} + T_{off}$ is too big, or after being used in "JETCALL3").

T1BITS: Value of jet pattern to be sent for the time duration specified by T_{on1} (contents loaded into NUJET for transmission via channel 6). These jets are required to go from present phase-plane point to the desired (or maximum) rate, where a coast interval (T_{off}) takes place and then the T2BITS jets are fired for T_{on2} to drive to the origin.

T2BITS: Single precision cell used to contain the value of the jet pattern to be sent for the time duration specified by T_{on2}: see discussion of T1BITS.

T5LOC, T6LOC: See Digital Autopilot Interface Routines.

TBITS: Value of the jet "pattern" to be sent for the time interval specified by T_{off}: it is set zero in "JETCALL1" (and used for symmetry in the program logic).

TIME5: See Digital Autopilot Interface Routines.

TUSED: Single precision cell, scale factor B14, units centi-seconds, used to keep track of the time elapsed by T_{on1}, T_{off}, and T_{on2}, in order to ensure that waitlist calls will not be set to elapse after the start of the next computing cycle (2 seconds) of the roll entry DAP, and also to make sure that minimum-firing time constraints on the jets are not violated. Cell is set to - TIME1 in "SETJTAG" (about 5 cs, plus computing delays, before the two-second entry roll DAP cycle is performed), and then is set to TIME1 + TUSED (a nominal 5 cs) before the start of processing, with an additional 2(8192) added to compensate, if necessary, for TIME1 overflow. This 5 centi-seconds, plus the two-second cycle, is then subtracted (giving nominally - 200 cs), to which 2 cs is added to avoid problems with the quantization of one cs for the waitlist and the fact that TUSED must exceed 0 for branches in equation logic to be taken: the 5, 200, and 2 are combined into the one constant K_{mtm3}. Additions of T_{on1}, T_{off}, and T_{on2} to TUSED (which starts at a nominal -198 cs) are tested for a positive non-zero result indicating expiration of the two-second computing interval.

VDTd180: Single precision cell, scale factor B-1, units revolutions/DAP cycle, giving the velocity argument for use in computing jet on and off times. Can be loaded with either drift velocity, maximum allowable velocity, or present velocity. If in deadzone, is loaded with 0 (for telemetry purposes).

VSQd4API: Single precision cell, scale factor B1, units revolutions, proportional to square of present "roll" body angular velocity divided by nominal acceleration ($K_{180d8att}$ mVTd180^2).

XDd360: Single precision value of predicted intercept of attitude error axis (rate = 0) by vehicle "trajectory", decremented by K_{xsd360}, scale factor B0, units revolutions.

ZREG: Hardware Z register (program counter), which can be used as a convenient source of a positive non-zero number (magnitude not significant). See QREG.

Digital Autopilot Interface Routines

T6RUPT   Entered based on program interrupt #1, controlled by TIME6

Proceed to address specified by T6LOC

T5RUPT   Entered based on program interrupt #2, controlled by TIME5

If TIME5 $\geqslant$ 50 cs, Resume (means TIME5 reset after interrupt was generated)

Proceed to address specified by T5LOC

HANDRUPT   Entered based on program interrupt #10, controlled by manual controller inputs

Resume

T5IDLOC

Resume   (if entered from TIME5 interrupt, TIME5 will cause an interrupt again in 2^{14} centi-seconds)

SETJTAG   Called by "READACCS" for entry DAP

TUSED = - TIME1

JETAG = 1

End of task

STABLISH   Entered for a V46E

If bits 15-14 (DAPBIT1, DAPBIT2) of FLAGWRD6 = 10_2: (TVC DAP)

If bit 14 of DAPDATR1 = 1:   (LM-on)

Perform "SWICHOVR"

Proceed to "PINBRNCH"

Proceed to "ALM/END"

Set bit 9(S4B Takeover Enable) of channel 12 = 0

TS = (bits 14-13 of DAPDATR1, shifted right 12 places) (to bits 2-1)

Inhibit interrupts

If TS = 00_2:   (no DAP)

Perform "ZEROJET"

HOLDFLAG = 1

(If TS = 00_2):

T5LOC = "T5IDLOC"

Set bits 15-14 (DAPBIT1, DAPBIT2) of FLAGWRD6 = 0

Proceed to "PINBRNCH"

If TS = 11_2: (Saturn DAP)

Proceed to "SATSTKON"

Perform "RCSDAPON" (TS = 01_2 or 10_2, for RCS DAP)

Proceed to "PINBRNCH"

DAPDISP Entered for a V48E (this is Routine 03)

If bits 15-14 (DAPBIT1, DAPBIT2) of FLAGWRD6 = 10_2: (TVC DAP)

Proceed to "ALM/END"

Perform "TESTXACT"

Establish "DONOUN46" (priority 10_8)

End of job

DONOUN46

TS = 0446_{vn}

Proceed to "GOXDSPF": if terminate, proceed to "ENDEXT"
if proceed, proceed
otherwise, proceed to previous line

If bit 4 of DAPDATR1 = 1: (maximum deadband specified)

Set bit 12(MAXDBFLG) of FLAGWRD9 = 1

If bit 4 of DAPDATR1 = 0:

Set bit 12(MAXDBFLG) of FLAGWRD9 = 0 (bit used in "INITSUB")

Perform "S41.2"

TS = 0647_{vn}

Proceed to "GOXDSPF": if terminate, proceed to "ENDEXT"
if proceed, proceed
otherwise, proceed to previous line

If bits 14-13 of DAPDATR1 $\neq 00_2$:

If bits 14-13 of DAPDATR1 $\neq 11_2$:

Inhibit interrupts (DAPDATR1 first digit = 1,2,5,6)

Perform "MASSPROP"

Release interrupts

Perform "S40.14"

TS = 0648_{vn}

Proceed to "GOXDSPF": if terminate, proceed to "ENDEXT"
if proceed, proceed
otherwise, proceed to previous line

Proceed to "ENDEXT"

SETMAXDB

ADB = K_{maxdb}

Set bit 4 of DAPDATR1 = 1

Return

SETMINDB

THETADX = CDU

ADB = K_{mindb}

Set bit 4 of DAPDATR1 = 0

Return

STICKCHK (Entered with TS set to control pattern)

PMANNDX = bits 2-1 of TS

YMANNDX = bits 4-3 of TS, shifted right 2 places (to bits 2-1)

RMANNDX = bits 6-5 of TS, shifted right 4 places (to bits 2-1)

Return (Error if bits 15-7 of TS non-zero)

SPSOFF   Entered from "AUTO37", "ENGINOFF", "V97E", and "V97T"

Inhibit interrupts

$T_{evt} = T_{now}$

Set bit 7(ENGONFLG) of FLAGWRD5 = 0

Set bit 13(SPS Engine On) of channel 11 = 0

Set bit 14(S4B Cutoff) of channel 12 = 1

If TVCPHASE $\neq$ -1:

If bits 15-14 (DAPBIT1, DAPBIT2) of FLAGWRD6 = 10_2:   (TVC DAP)

If |REPFRAC| $\neq$ 0:

PACTOFF = DELPBAR_{sp}

YACTOFF = DELYBAR_{sp}

Return

TVCZAP

Set bits 11(Disengage Optics DAC), 8(TVC Enable), and 2(Enable Optics CDU Error Counters) of channel 12 = 0

OPTIND = -1   ("DAC" is digital-to-analog converter)

NVWORD1 = +0

Set bit 11(TIMRFLAG) of FLAGWRD7 = 0

Return

PRE40.6   Called by "V97E" or restart group 6.2

If bits 15-14(DAPBIT1, DAPBIT2) of FLAGWRD6 = 10_2:   (TVC DAP)

End of task

MRKRTMP = -1

CNTR = +1   (serves as flag that entered "S40.6" from here)

Proceed to "S40.6"

S40.6   Called by "P40SXTY"

OPTIND = -0

Set bit 2(Enable Optics CDU Error Counters) of channel 12 = 0

Set bits 11 (Disengage Optics DAC) and 8(TVC Enable) of channel 12 = 1

Delay 0.06 seconds

Set bit 2(Enable Optics CDU Error Counters) of channel 12 = 1

Delay 0.02 seconds

If CNTR > 0: (i.e. from "PRE40.6")

> TVCPITCH = PACTOFF - 0 (avoid loading with +0)
>
> TVCYAW = YACTOFF - 0 (avoid loading with +0)
>
> Set bits 12-11 (Gate output from TVCPITCH & TVCYAW) of channel 14 = 1
>
> End of task

If MRKRTMP > 0:

> Perform the following for i = PITCH and then i = YAW, with j = 11, 12 respectively:
>
> > TVCi = $K_{p2actdg}$
> >
> > Set bit j (Gate output from TVCi) of channel 14 = 1
> >
> > Delay 2 seconds
> >
> > TVCi = $K_{m4actdg}$
> >
> > Set bit j (Gate output from TVCi) of channel 14 = 1
> >
> > Delay 2 seconds
> >
> > TVCi = $K_{p2actdg}$
> >
> > Set bit j (Gate output from TVCi) of channel 14 = 1
> >
> > Delay 2 seconds

Delay 4 seconds

TVCPITCH = PACTOFF - 0

TVCYAW = YACTOFF - 0

Set bits 12-11 (Gate outputs from TVCPITCH & TVCYAW) of channel 14 = 1

End of task

S40.14 Entered from "DONOUN46" and "REDAP"

$JdM_0 = K_{cntone}$ IXX

$JdM_1 = K_{cntone}$ IAVG

$JdM_2 = JdM_1$

$KMJ_0 = (K_{cnttwo})$ / IXX

$KMJ_1 = (K_{cnttwo})$ / IAVG

$KMJ_2 = KMJ_1$

Return

S40.15 Entered from "SWICHOVR", "TVCEXEC", and "TVCINIT1"

1dCONACC = K_{2pidm} IXX

VARK = KTLXdI IAVGdTLX

Return

S41.2 Entered from "DONOUN46" and "REDAP"

RATEINDX = 2 (bits 2-1 of DAPDATR1)

Inhibit interrupts

If bits 14-13 of DAPDATR1 $\neq 10_2$:

Set bit 2(LMATTCH) of FLAGWRD7 = 0

If bits 14-13 of DAPDATR1 = 10_2:

Set bit 2(LMATTCH) of FLAGWRD7 = 1

Release interrupts

TS = K_{dc46}

If bit 4 of DAPDATR1 = 1: (use wide deadband)

TS = TS + K_{dc409}

ADB = TS

If bit 7 of DAPDATR1 = 1: (use B/D for X-translation)

XTRANS = 1

If bit 7 of DAPDATR1 = 0:

XTRANS = 0

If bit 10 of DAPDATR1 = 1: (use A/C for X-translation)

XTRANS = XTRANS - 1

If XTRANS = 0:

Set bit 15(2JETSFLG) of FLAGWRD1 = 0

If XTRANS $\neq$ 0:

Set bit 15(2JETSFLG) of FLAGWRD1 = 1

If bit 13 of DAPDATR2 = 0: (use B/D for roll)

ACORBD = 1

If bit 13 of DAPDATR2 = 1: (use A/C for roll)

ACORBD = - 4096 (i.e. negative non-zero)

If bit 10 of DAPDATR2 = 1: (Quad A OK)

RACFAIL = 0

If bit 4 of DAPDATR2 = 0: (Quad C not OK)

RACFAIL = -1

If bit 10 of DAPDATR2 = 0: (Quad A not OK)

RACFAIL = 1

If bit 7 of DAPDATR2 = 1: (Quad B OK)

RBDFAIL = 0

If bit 1 of DAPDATR2 = 0: (Quad D not OK)

RBDFAIL = -1

If bit 7 of DAPDATR2 = 0: (Quad B not OK)

RBDFAIL = 1

Return

NEEDLER

If bit 4 (Coarse Align) of channel 12 = 1:

Set bit 3 of RCSFLAGS = 1

Return

If bit 3 of RCSFLAGS = 1:

Set bit 6 (Enable CDU IMU Error Counters) of channel 12 = 0

Proceed to "NEEDLE11"

If bit 2 of RCSFLAGS = 1, proceed to "NEEDLER2"

If bit 6 (Enable CDU IMU Error Counters) of channel 12 = 0:

Set bit 3 of RCSFLAGS = 1

Return

Proceed to "NEEDLES"

NEEDLE11

$A\underline{K} = -0$

$EDRIV\underline{E} = -0$

$CDUXCM\underline{D} = -0$

Set bits 3-2 of RCSFLAGS = 01_2

Return

NEEDLER2

Set bit 6 (Enable CDU IMU Error Counters) of channel 12 = 1

Set bits 3-2 of RCSFLAGS = 00_2

Return

NEEDLES

$T\underline{S} = -K_{dacsc}\ A\underline{K}$, with magnitude of each component limited $\leq K_{dacmit}$

$CDUXCM\underline{D} = CDUXCM\underline{D} + (T\underline{S} - EDRIV\underline{E})$

$EDRIV\underline{E} = T\underline{S}$

Set bits 15-13 (Gate outputs from $CDUXCM\underline{D}$) of channel 14 = 1

Return

MASSPROP Entered from "AUTO37", "DONOUN46", "ENGINOFF", and "TVCINIT1"

If bit 13 of DAPDATR1 = 1: (means LM-off)

$\text{VARSTi} = K_{\text{nolemval}_i}$ (i = 0 - 9)

If bit 13 of DAPDATR1 = 0:

$\text{VARSTi} = K_{\text{intvalue}_i} + K_{\text{slopeval}_i}\ \text{LEMMASS}$ (i = 0 - 9)

If bit 15 of DAPDATR1 = 1: (descent stage off)

VARST1 = VARST1 + VARST8

VARST2 = VARST2 + VARST9

(VARST1 and VARST2 not incremented if DAPDATR1 is 77777_8, which it should not)

$\text{VARST7} = \text{VARST7} + K_{\text{dxitfix}}$ (VARST8 and VARST9 destroyed if used)

Proceed to "FIXCW"

FIXCW Entered from "TVCEXEC" and at end of "MASSPROP"

$\text{DELCMWT} = \text{CSMMASS} + K_{\text{negbpw}}$

If DELCMWT $\leq$ 0:

IAVGdTLX = VARST7 DELCMWT + VARST2

IAVG = VARST6 DELCMWT + VARST1

IXX = VARST5 DELCMWT + VARST0

If DELCMWT > 0:

IAVGdTLX = VARST3 DELCMWT + VARST2

IAVG = VARST4 DELCMWT + VARST1

IXX = VARST5 DELCMWT + VARST0

TS = CSMMASS

If bit 14 of DAPDATR1 = 1: (means LM-on)

TS = TS + LEMMASS

MASS = TS

Return (to routine calling "MASSPROP" or "FIXCW")

Quantities in Computations

See also list of major variables and list of routines

1dCONACC: Single precision value of reciprocal of roll-axis acceleration used for TVC roll DAP, scale factor B9, units of $(\text{revolutions/second}^2)^{-1}$. Computed in "S40.15", and is based on two-jet torque (see K_{2pidm}). Updated every 10 seconds during burn (by entrance to "S40.15").

ACORBD: Single precision quantity, scale factor B14, specifying the quad pair to be used for roll attitude control in the RCS DAP. If positive (+1), quad B/D is to be used; if negative (-4096, for programming convenience), quad A/C is used. The TVC DAP alternates roll commands between quads. Cell is set in "S41.2" based upon bit 13 of DAPDATR2 (0 for positive, 1 for negative).

ADB: Single precision value of the attitude error deadband used in RCS DAP, scale factor B-1, units revolutions (one least increment is about 0.011°). It is set initially based on bit 4 of DAPDATR1 (about 0.5° for 0, about 5° for 1), but can subsequently be changed by the program (with changes reflected in DAPDATR1 for transfers between "wide" and "narrow" deadbands). In P20, it is set based on R2 of N79, except that 0 there is interpreted to be 0.5°, each pass through "R61CSM" (R61CNTR = 0) or in "R67START". The deadband is restored to the R03 value (using bit 12 of FLAGWRD9) in "FIXDB" and "INITSUBA". The R03 value is retained when "DONOUN46" is performed; a noun 01 procedure could also be used to load the cell, if desired.

$\underline{AK}$ (AK_0, AK_1, AK_2): Single precision values of attitude errors used as communication cells with "NEEDLER", scale factor B-1, units revolutions. Nominal scaling of display (see K_{dacsc}) is such that "full scale deflection" is 16.875° of attitude error, with a positive $\underline{AK}$ input to "NEEDLER" giving a negative loading of $\underline{\text{CDUXCMD}}$.

During P11, entered from "NOPOLYM" with AK_0 loaded so as to give full scale of 67.5° (other axes normal scaling), every half second (approximately). Can be used to steer Saturn if control of Saturn switch set. If Saturn DAP activated, entered from "SATSTICK" instead.

During RCS DAP operation, "NEEDLER" is entered every 0.2 seconds from "KMATRIX" (on alternate 0.2 seconds, $\underline{AK}$ is loaded in that routine with appropriate error information).

During TVC DAP operation, "NEEDLER" is entered every 0.5 second from "TVCEXEC", with $\underline{AK}$ loaded by "TVCEXEC", "PCOPY", and "YCOPY" respectively.

During entry DAP operation, "NEEDLER" is entered from "CM/FDAIR" every 0.2 second, with AK_0 loaded so as to give full scale of 67.5° and others having normal scaling. AK_0 is set 0 every 2 seconds if CMDAPMOD = -0 (see Digital Autopilot Entry Routines), and AK_1 and AK_2 effectively set 0 if bit 3 (05GSW) of FLAGWRD6 = 1: they are otherwise computed in "ATTRATES" and "EXDAP" respectively.

The general polarity of loading $\underline{AK}$ is "commanded angle minus present angle". A DSKY one-shot drive of error needles can be done by V43E means (from "ATTCK1").

CDUXCMD: See IMU Computations. If coarse align bit is not set, the data goes to FDAI attitude error display (x, y, z associated with roll, pitch, and yaw respectively, with 384 pulses giving full scale deflection). If bit 9 of channel 12 = 1, the information in the cells is used for Saturn control.

CNTR: Single precision cell used for switch-over control purposes in TVC DAP. Prior to entering TVC DAP, is used in "S40.6" as a flag to indicate whether that routine entered from "PRE40.6" (value 1, scale factor B14) or from "P40SXTY" (value +0). If it was entered from "PRE40.6", then the four-second delay before driving the SPS gimbals to their trim position is bypassed.

CSMMASS: Value of CSM mass, scale factor B16, units kilograms, computed double precision in "S40.8" but used single precision elsewhere (including DSKY loading). Since the least significant half is not otherwise employed, the displayed value can differ from the onboard estimate by $\pm$~4 kg (about 8.8 pounds). It can be loaded manually in R1 of N47, and is updated in "S40.8". The updating consists of subtracting two seconds worth of mass flow rate (determined by a pad-load constant in units of kilograms/centi-second) if the sensed accelerometer output in the previous two-second interval exceeded the thrust-fail criterion (another pad-load constant). The updating occurs whenever "S40.8" is entered (and the accelerometer output is sufficient), and uses MASSTMP (see "S40.8") for restart protection purposes. "S40.8" is entered only for P40, starting when Average-G is turned on (due to AVEGEXIT setting in "P40SXTY"), and ending when AVEGEXIT is loaded with some other quantity (e.g. "CALCN85" in "P40RCS" for an ENTR to F V99 or PRO to F V16 N40 after the burn). Because of this logic, no other special provisions need be made to compensate for thrust fail conditions: it should be noted, however, that the gain updates in "TVCEXEC" are based on the current value of CSMMASS. Although the mass updating is done as part of the Average-G computation loop, the computations are entered <u>only</u> in P40. As a result, if an SPS burn were to be done using the SCS and P47, for example, CSMMASS would have to be updated manually.

DAPDATR1: Single precision quantity whose individual octal digits are assigned control significance for the RCS and TVC DAPs, and which is displayed in R1 of N46. Digit #1 is used for the TVC DAP to determine vehicle configuration (LM-off, LM-on descent & ascent, or LM-on ascent only), while the remaining digits are used only by the RCS DAP (together with digit #1). Digits #2 - #5 are processed in "S41.2", which is entered when R03 is performed as well as when the RCS DAP is initialized (e.g. after a restart or when IMU data becomes usable after a period of not being usable). The individual bits of DAPDATR1 have the following meanings(digit #1 is also referred to as "A", #2 as "B", etc.):

Octal Digit	Word Bit	Significance
1	15-13	Specify vehicle configuration/desired DAP. Because of the use of various bits (e.g. 14 and 13) at different parts of the program to obtain vehicle information, caution should be used that the digits specified be those below, and other patterns should be avoided. 0 is set by "CM/DAPON" for entry DAP, and terminates other DAP activity if "STABLISH" is entered 1 indicates LM-off (causes RCS DAP to be started if "STABLISH" is entered) 2 indicates LM-on ascent and descent, and causes RCS DAP to be started if "STABLISH" is entered 3 indicates Saturn DAP, and causes Saturn DAP to be started if "STABLISH" is entered 6 indicates LM-on ascent only, and causes RCS DAP to be started if "STABLISH" is entered (bit 15 is used in "MASSPROP" to select proper coefficient values)
2	12-11	Not used (no effect)
	10	Allow use of quad A/C for X-translation if 1
3	9-8	Not used (no effect)
	7	Allow use of quad B/D for X-translation if 1 (if bits 10 and 7 both 0, program acts as if both bits were 1)
4	6-5	Not used (no effect)
	4	Select narrow deadband (0.5°) if 0 and wide deadband (5.0°) if 1: loading of ADB is done

Octal Digit	Word Bit	Significance
5	3	Not used (no effect)
	2-1	Specify maneuver rate for RHC or automatic maneuvers: 00_2 for 0.05°/sec; 01_2 for 0.20°/sec; 10_2 for 0.50°/sec; 11_2 for 2.00°/sec

DAPDATR2: Single precision quantity whose individual octal digits are assigned control significance for the RCS DAP only, and which is displayed in R2 of N46. It is processed in "S41.2", with the individual bits having the meanings given below:

Octal Digit	Word Bit	Significance
1	15-14	Not used (no effect)
	13	Use quad A/C for roll if 1 (if 0, use B/D)
2	12-11	Not used (no effect)
	10	Quad A may be used if 1 (not to be used if 0)
3	9-8	Not used (no effect)
	7	Quad B may be used if 1 (not to be used if 0)
4	6-5	Not used (no effect)
	4	Quad C may be used if 1 (if bit 10 is 1, quad C not to be used if bit 4 is 0)
5	3-2	Not used (no effect)
	1	Quad D may be used if 1 (if bit 7 is 1, quad D not to be used if bit 1 is 0)

DELCMWT: Value of (CSMMASS + K_{negbpw}) computed in "FIXCW", scale factor B15, units kilograms, single precision (the difference between present CSM mass and the "breakpoint" value for use in computing mass-property information). Program notation is "TEMP333".

DELPBAR, DELYBAR: See Digital Autopilot TVC Routines.

EDRIV$\underline{E}$: Single precision values of previous scaled A$\underline{K}$ output information in "NEEDLES", scale factor B14, units output drive pulses (there are 2^{13} pulses per revolution for standard scaling). EDRIV$\underline{E}$ is the internal computer image of the number contained in the appropriate error counter driven from CDUXCM$\underline{D}$, and is required since interface is incremental pulses from a previous setting.

HOLDFLAG: Single precision quantity, scale factor B14, used to control the nature of the attitude hold performed by the RCS DAP. A positive non-zero value means that at the next opportunity ("AHFNOROT" entered not in Free Mode and with rate filter initialized), the present CDU angles should be loaded into THETAD$\underline{X}$ for use as the new attitude reference. A +0 value means that the DAP is in attitude hold about the angles in THETAD$\underline{X}$. A negative non-zero value means that automatic maneuvers are being performed (and WBOD$\underline{Y}$, CDUX$\underline{D}$, BIA$\underline{S}$, and DELCD$\underline{U}$ all have an effect, see Digital Autopilot RCS Routines).
A positive non-zero value is loaded if the DAP's control of the vehicle is removed (e.g. bit 15 of channel 31 = 1); when the DAP is initialized; if the IMU data is not usable (i.e. not fine align); if the Free Mode is entered; if RHC commands are provided; or if no DAP is desired (by "STABLISH"). If HOLDFLAG is negative non-zero and a switch to Hold Mode is made, attitude hold is also performed (same effect as having HOLDFLAG positive non-zero, although the setting is not made).
A positive zero value is set in "AHFNOROT" after loading THETAD$\underline{X}$ provided that bits 13-11 of RCSFLAGS = 0 (i.e. damping done), and in "T5PHASE2" if the magnitude of the middle gimbal angle exceeds 75°.
A negative non-zero value is set in R61 and also when attitude maneuver computations (e.g. R60 or R67) have loaded the DAP interface cells. Programs checking HOLDFLAG can conclude, if it is not negative non-zero, that an interruption of the automatic maneuver has occurred.

IAVG: Single precision value of the "average" of moments of inertia about the y and z axes, scale factor B20, units kilogram-meters^2. It is computed as a function of CSMMASS (and LEMMASS if DAPDATR1 so indicates) in "FIXCW" and is used in "S40.14" to compute quantities for the RCS DAP (the separate quantity IAVGdTLX is used for the TVC DAP). Storage address corresponds to "IXX" +1.

IAVGdTLX: Single precision value of the "average" of moments of inertia about the y and z axes divided by the "thrust moment" (product of engine thrust and the "moment arm from hinge-point to c.g."), scale factor B2, units seconds^2, used in "S40.15" to multiply KTLXdI to obtain VARK. Program notation is "IAVG/TLX", and storage address corresponds to "IXX" +2. Computed in "FIXCW".

IXX: Single precision value of moment of inertia about the x axis (roll), scale factor B20, units kilogram-meters^2, computed in "FIXCW". It is used in "S40.14" and "S40.15" to compute quantities for the RCS and TVC DAPs respectively.

Jd$\underline{M}$ (JdM_0, JdM_1, JdM_2): Single precision value of 80% of the reciprocal of one-jet acceleration about roll, pitch, and yaw axes respectively, scale factor B23, units of (revolutions/(deci-second centi-second))$^{-1}$, so that multiplication of a rate in revolutions/deci-second by Jd$\underline{M}$ yields a time in centi-seconds. The 80% factor is in K_{cntone}, to account for "tolerances in the torque produced by the jets and uncertainties in the vehicle moment of inertia". The quantities are computed in "S40.14", with pitch and yaw values the same.

JETAG: See Digital Autopilot Entry Routines.

K_{2pidm}: Single precision constant, program notation "2PI/M", scale factor B-8. Nominal value is 0.00331017×2^8, corresponding to the reciprocal of: $(700 \times 2 \times 1.355817948 \times (1/2\pi) \times 2^{-8})$, where first term is nominal torque due to a single jet (in foot-pounds), second accounts for two jets, third converts to units of newton-meters, fourth converts to units of revolutions, and fifth is scale factor.

K_{cntone}: Single precision constant, program notation "CONTONE", scale factor B3, value 0.662034. Value corresponds to the reciprocal of: $(700 \times 1.355817948 \times (1/2\pi) \times (1/0.8) \times 10^{-2} \times 10^{-1} \times 2^3)$, where first term is nominal torque due to single jet (in foot-pounds), second converts to units of newton-meters, third converts to units of revolutions, fourth is the 80% factor discussed for Jd$\underline{M}$, fifth converts to units of centi-seconds for result, sixth converts rotation rate to units of deci-seconds (see Jd$\underline{M}$ for the time units), and seventh is the scale factor.

K_{cnttwo}: Constant, program notation "CONTTWO", scale factor B7, value 0.00118 (double precision). Value corresponds to $700 \times 1.355817948 \times (1/2\pi) \times 10^{-1} \times 10^{-2} \times 2^{-7}$, where first term is nominal torque due to single jet (in foot-pounds), second converts to units of newton-meters, third converts to units of revolutions, fourth converts to deci-seconds (for rotation rate), fifth converts to centi-seconds (see KM$\underline{J}$ for time units), and sixth is scale factor.

K_{dacmit}: Single precision constant, program notation "DACLIMIT +1", scale factor B14, units of output drive pulses. Value is 384×2^{-14}, corresponding to 384 pulses (the saturated error counter). Actual program implementation involves 3 constants: -384, 16000, and +384, with the 16000 used as a magnitude check. If, when 16000 is added to magnitude of argument, the answer overflows (exceeds 16383), then limiting is required. The 384 corresponds to 600_8.

K_{dacsc}: Single precision constant, program notation "QUARTER", scale factor B15, value $\frac{1}{4}$, corresponding to a true value of 2^{13}. It converts A$\underline{K}$ from units of revolutions to units of output drive pulses, with 2^{13} pulses corresponding to the nominal scaling for a revolution. Hence 384 pulses, the error counter saturation level, correspond to 16 7/8 degrees, as discussed for A$\underline{K}$.

K_{dc46}: Single precision constant, program notation "DEC46", scale factor B-1, units revolutions. Value is 46×2^{-14}, the same as K_{mindb}.

K_{dc409}: Single precision constant, program notation "DEC409", scale factor B-1, units revolutions. Value is 409×2^{-14}, which when added to K_{dc46} gives 455×2^{-14}, the same as K_{maxdb}.

$K_{dxitfix}$: Single precision constant, program notation "DXITFIX", scale factor B-12, units seconds^2/kilogram, used to increment VARST7 if LM-on APS-only configuration is specified. Nominal value is $-1.88275E-5 \times 2^{12}$, corresponding to $-0.854E-5 \times (1/0.45359237) \times 2^{12}$, where first term is value in sec^2/pound, second converts to kilograms, and third is scale factor. Octal value is 75420_8, corresponding to about -0.85367E-5 sec^2/pound.

$K_{intvalue_0}$: Single precision constant, program notation "INTVALUE", scale factor B20, units kilogram-meters^2, used as the constant value to initialize VARST0 for LM-on. Nominal value is 26850×2^{-20}, corresponding to about $19803.5 \times 1.355817948 \times 2^{-20}$, where first term is value in slug-ft^2, second converts to kilogram-meters^2, and third is scale factor. Octal value is 00644_8, corresponding to 26880 kilogram-meters^2 or about 19826 slug-ft^2.

$K_{intvalue_1}$: Single precision constant, program notation "INTVALUE +1", scale factor B20, units kilogram-meters^2, used as the constant value to initialize VARST1 for LM-on (if descent stage off, subsequently incremented by VARST8). Nominal value is 127518×2^{-20}, corresponding to about $94052.5 \times 1.355817948 \times 2^{-20}$, where first term is value in slug-ft^2. Octal value is 3710_8, corresponding to 127488 kilogram-meters^2 or about 94030 slug-ft^2.

$K_{intvalue_2}$: Single precision constant, program notation "INTVALUE +2", scale factor B2, used to initialize VARST2 for LM-on (if descent stage off, subsequently incremented by VARST8) by providing the constant value. Nominal value is 0.54059×2^{-2}, corresponding to 0.54059 seconds^2 (since units of constant are seconds^2). Octal value is 4246_8, corresponding to about 0.540527 seconds^2.

$K_{intvalue_3}$: Single precision constant, program notation "INTVALUE +3", scale factor B-12, used to initialize VARST3 for LM-on, by providing the constant value. Nominal value is $0.153964E-4 \times 2^{12}$, with units seconds^2/kilogram. Nominal value corresponds to about $0.698369E-5 \times (1/0.45359237) \times 2^{12}$, where first term is value in seconds^2/pound, second converts to kilograms, and third is scale factor. Octal value is 2011_8, corresponding to about 0.698210E-5 seconds^2/pound.

$K_{intvalue_4}$: Single precision constant, program notation "INTVALUE +4", scale factor B6, used to initialize VARST4 for LM-on, by providing the constant value, units (kilogram-meters^2)/kilogram. Nominal value is -0.742923×2^{-6}, corresponding to about $-0.2485468 \times 1.355817948 \times (1/0.45359237) \times 2^{-6}$, where first term is value in slug-ft^2/pound, second converts to kilogram-meters^2, third converts to kilograms, and fourth is scale factor. Octal value is 77501_8, corresponding to about -0.24830 slug-ft^2/pound.

$K_{intvalue_5}$: Single precision constant, program notation "INTVALUE +5", scale factor B6, used to initialize VARST5 for LM-on, by providing the constant value, units (kilogram-meters^2)/kilogram. Nominal value is 1.5398×2^{-6}, corresponding to about $0.51514 \times 1.355817948 \times (1/0.45359237) \times 2^{-6}$, where first term is value in slug-ft^2/pound. Octal value is 612_8, corresponding to about 0.5149 slug-ft^2/pound. Octal value same as $K_{nolemval_5}$.

$K_{intvalue_6}$: Single precision constant, program notation "INTVALUE +6", scale factor B6, used to initialize VARST6 for LM-on, by providing the constant value, units (kilogram-meters^2)/kilogram. Nominal value is 9.68×2^{-6}, corresponding to about $3.2385 \times 1.355817948 \times (1/0.45359237) \times 2^{-6}$, where first term is value in slug-ft^2/pound. Octal value is 4656_8, corresponding to about 3.2384 slug-ft^2/pound.

$K_{intvalue_7}$: Single precision constant, program notation "INTVALUE +7", scale factor B-12, used to initialize VARST7 for LM-on, by providing the constant value, units seconds^2/kilogram. Nominal value is $0.647625\text{E-}4 \times 2^{12}$, corresponding to about $0.293758\text{E-}4 \times (1/0.45359237) \times 2^{12}$, where first term is value in sec^2/pound, second converts to kilograms, and third is scale factor. Octal value is 10372_8, corresponding to about 0.293748E-4 sec^2/pound.

$K_{intvalue_8}$: Single precision constant, program notation "INTVALUE +8", scale factor B20, used to initialize VARST8 for LM-on, by providing the constant value, units kilogram-meters^2. Nominal value is -27228×2^{-20}, corresponding to about $-20082 \times 1.355817948 \times 2^{-20}$, where first term is value in slug-ft^2, second converts to kilogram-meters^2, and third is scale factor. Octal value is 77126_8, corresponding to -27200 kilogram-meters^2 or about -20062 slug-ft^2.

$K_{intvalue_9}$: Single precision constant, program notation "INTVALUE +9", scale factor B2, used to initialize VARST9 for LM-on, by providing the constant value, units seconds^2. Nominal value is -0.206476×2^{-2}, corresponding to -0.206476 seconds^2. Octal value of constant is 76261_8, corresponding to about -0.206543 seconds^2.

$K_{m4actdg}$: Single precision constant, program notation "-4ACTDEG", scale factor B14, units CDU actuator output pulses (one pulse is 85.41 seconds; there are about 42.14963 pulses/degree). Value is -168×2^{-14}, or twice the value of $K_{p2actdg}$ (with opposite sign), corresponding to about -4 degrees.

K_{maxdb}: Single precision constant, program notation "MAXDB", scale factor B-1, units revolutions. Value is 455×2^{-14}, corresponding to about 4.9988 degrees.

K_{mindb}: Single precision constant, program notation "MINDB", scale factor B-1, units revolutions. Value is 46×2^{-14}, corresponding to about 0.5054 degrees.

K_{negbpw}: Single precision constant, program notation "NEGBPW", scale factor B16, units kilograms, giving the complement of the CSM "breakpoint" weight for computation of mass properties. Nominal value is -15402.17×2^{-16}, corresponding to -15402.17 kilograms or about 33955.97 pounds (program comments indicate that "dry weight" is 23956 pounds, giving effect for constant of 10,000 pounds of propellant). Octal value is 70364_8, corresponding to -15404 kilograms (-33960.0 pounds).

$K_{nolemval_0}$: Single precision constant, program notation "NOLEMVAL", scale factor B20, units kilogram-meters^2, giving setting for VARSTO for LM-off. Nominal value is 25445×2^{-20}, corresponding to about $18767 \times 1.355817948 \times 2^{-20}$, where first term is value in slug-ft^2, second converts to kilogram-meters^2, and third is scale factor. Octal value is 616_8, corresponding to about 18787 slug-ft^2.

$K_{nolemval_1}$: Single precision constant, program notation "NOLEMVAL +1", scale factor B20, units kilogram-meters^2, used to give setting for VARST1 for LM-off. Nominal value is 87450×2^{-20}, corresponding to about $64500 \times 1.355817948 \times 2^{-20}$, where first term is value in slug-ft^2. Octal value is 2526_8, corresponding to about 64481 slug-ft^2.

$K_{nolemval_2}$: Single precision constant, program notation "NOLEMVAL +2", scale factor B2, used to give setting for VARST2 for LM-off, units seconds^2. Nominal value is 0.30715×2^{-2}, corresponding to 0.30715 seconds^2. Octal value is 2352_8, corresponding to about 0.30713 seconds^2.

$K_{nolemval_3}$: Single precision constant, program notation "NOLEMVAL +3", scale factor B-12, units seconds^2/kilogram, used to initialize VARST3 for LM-off. Nominal value is $1.22877E-5 \times 2^{12}$, corresponding to $0.55736E-5 \times (1/0.45359237) \times 2^{12}$, where first term is value in seconds^2/pound, second converts to kilograms, and third is scale factor. Octal value is 1471_8, corresponding to about 0.55762E-5 seconds^2/pound.

$K_{nolemval_4}$: Single precision constant, program notation "NOLEMVAL +4", scale factor B6, used to initialize VARST4 for LM-off, units kilogram-meters^2 per kilogram. Nominal value is 1.6096×2^{-6}, corresponding to about $0.5385 \times 1.355817948 \times (1/0.45359237) \times 2^{-6}$, where first term is value in slug-ft^2/pound, second converts to kilogram-meters^2, third converts to kilograms, and fourth is scale factor. Octal value is 634_8, corresponding to about 0.53842 slug-ft^2/pound.

$K_{nolemval_5}$: Single precision constant, program notation "NOLEMVAL +5", scale factor B6, used to initialize VARST5 for LM-off, units kilogram-meters^2 per kilogram. Nominal value is 1.54×2^{-6}, corresponding to about $0.5152 \times 1.355817948 \times (1/0.45359237) \times 2^{-6}$, where first term is value in slug-ft^2/pound, second converts to kilogram-meters^2, third converts to kilograms, and fourth is scale factor. Octal value is 612_8, corresponding to about 0.5149 slug-ft^2/pound. Octal value same as $K_{intvalue_5}$.

$K_{nolemval_6}$: Single precision constant, program notation "NOLEMVAL +6", scale factor B6, used to initialize VARST6 for LM-off, units kilogram-meters^2 per kilogram. Nominal value is 7.77177×2^{-6}, corresponding to about $2.600 \times 1.355817948 \times (1/0.45359237) \times 2^{-6}$, where first term is value in slug-ft^2/pound, second converts to kilogram-meters^2, third converts to kilograms, and fourth is scale factor. Octal value is 3706_8, corresponding to about 2.6006 slug-ft^2/pound.

$K_{nolemval_7}$: Single precision constant, program notation "NOLEMVAL +7", scale factor B-12, used to initialize VARST7 for LM-off, units seconds^2/kilogram. Nominal value is $3.46458E-5 \times 2^{12}$, corresponding to about $1.5715E-5 \times (1/0.45359237) \times 2^{12}$, where first term is value in sec^2/pound, second converts to kilograms, and third is scale factor. Octal value is 4425_8, corresponding to about 1.57148E-5 sec^2/pound.

$K_{nolemval_8}$: Single precision constant loaded into VARST8 for LM-off, but should not be used. Cell corresponds to $K_{intvalue_0}$, with same scaling.

$K_{nolemval_9}$: Single precision constant loaded into VARST9 for LM-off, but should not be used. Cell corresponds to $K_{intvalue_1}$, but with octal number considered scaled B2 instead of B20. Value would be about 0.48633 seconds^2.

$K_{p2actdg}$: Single precision constant, program notation "+2ACTDEG", scale factor B14, units CDU actuator pulses (see $K_{m4actdg}$). Value is 84×2^{-14}, corresponding to about +2 degrees.

$K_{slopeval_0}$: Single precision constant, program notation "SLOPEVAL", scale factor B6, units kilogram-meters^2 per kilogram, used as the slope value to initialize VARST0 for LM-on. Nominal value is 1.96307×2^{-6}, corresponding to about $0.65675 \times 1.355817948 \times (1/0.45359237) \times 2^{-6}$, where first term is value in slug-ft^2/pound, second converts to kilogram-meters^2, third converts to kilograms, and fourth is scale factor. Octal value is 767_8, corresponding to about 0.65734 slug-ft^2/pound.

$K_{slopeval_1}$: Single precision constant, program notation "SLOPEVAL +1", scale factor B6, units kilogram-meters^2/kilogram, used as the slope value to initialize VARST1 for LM-on. Nominal value is 27.5774×2^{-6}, corresponding to about $9.2261 \times 1.355817948 \times (1/0.45359237) \times 2^{-6}$, where first term is value in slug-ft^2/pound. Octal value is 15624_8, corresponding to about 9.2263 slug-ft^2/pound.

$K_{slopeval_2}$: Single precision constant, program notation "SLOPEVAL +2", scale factor B-12, units seconds^2/kilogram, used as the slope value to initialize VARST2 for LM-on. Nominal value is $2.3548\text{E-}5 \times 2^{12}$, corresponding to about $1.06812\text{E-}5 \times (1/0.45359237) \times 2^{12}$, where first term is value in seconds^2/pound, second converts to kilograms, and third is scale factor. Octal value is 3054_8, corresponding to about 1.06793E-5 seconds^2/pound.

$K_{slopeval_3}$: Single precision constant, program notation "SLOPEVAL +3", scale factor B-26, units seconds^2/kilogram per kilogram, used as the slope value to initialize VARST3 for LM-on. Nominal value is $2.1777\text{E-}9 \times 2^{26}$, corresponding to about $0.44805\text{E-}9 \times (1/0.45359237)^2 \times 2^{26}$, where first term is value in seconds^2/pound per pound. Octal value is 4532_8, corresponding to about 0.44798E-9 seconds^2/pound per pound.

$K_{slopeval_4}$: Single precision constant, program notation "SLOPEVAL +4", scale factor B-8, units kilogram-meters^2 per kilogram per kilogram, used as the slope value to initialize VARST4 for LM-on. Nominal value is $1.044\text{E-}3 \times 2^8$, corresponding to about $0.1584\text{E-}3 \times 1.355817948 \times (1/0.45359237)^2 \times 2^8$, where first term is value in slug-ft^2/pound per pound, second converts to kilogram-meters^2, third converts to kilograms, and fourth is scale factor. Octal value is 10433_8, corresponding to about 0.158433E-3 slug-ft^2/pound per pound.

$K_{slopeval_5}$: Single precision constant, program notation "SLOPEVAL +5", scale factor B-8, units kilogram-meters^2 per kilogram per kilogram, used as the slope value to initialize VARST5 for LM-on. Value is zero, meaning that VARST5 is not affected by LEMMASS.

$K_{slopeval_6}$: Single precision constant, program notation "SLOPEVAL +6", scale factor B-8, units kilogram-meters^2 per kilogram per kilogram, used as the slope value to initialize VARST6 for LM-on. Nominal value is $2.21068E-3 \times 2^8$, corresponding to about $0.335472E-3 \times 1.355817948 \times (1/0.45359237)^2 \times 2^8$, where first term is value in slug-ft^2/pound per pound. Octal value is 22070_8, corresponding to about 0.33546E-3 slug-ft^2/pound per pound.

$K_{slopeval_7}$: Single precision constant, program notation "SLOPEVAL +7", scale factor B-26, units seconds^2/kilogram per kilogram, used as the slope value to initialize VARST7 for LM-on. Nominal value is $1.5166E-9 \times 2^{26}$, corresponding to about $0.31203E-9 \times (1/0.45359237)^2 \times 2^{26}$, where first term is value in seconds^2/pound per pound, second converts to kilograms, and third is scale factor. Octal value is 3204_8, corresponding to about 0.31212E-9 seconds^2/pound per pound.

$K_{slopeval_8}$: Single precision constant, program notation "SLOPEVAL +8", scale factor B6, units kilogram-meters^2 per kilogram, used as the slope value to initialize VARST8 for LM-on. Nominal value is -1.284×2^{-6}, corresponding to about $-0.4296 \times 1.355817948 \times (1/0.45359237) \times 2^{-6}$, where first term is value in slug-ft^2/pound, second converts to kilogram-meters^2, third converts to kilograms, and fourth is scale factor. Octal value is 77266_8, corresponding to about -0.4300 slug-ft^2/pound.

$K_{slopeval_9}$: Single precision constant, program notation "SLOPEVAL +9", scale factor B-12, units seconds^2/kilogram, used as the slope value to initialize VARST9 for LM-on. Nominal value is $2.00E-5 \times 2^{12}$, corresponding to about $0.9072E-5 \times (1/0.45359237) \times 2^{12}$, where first term is value in seconds^2/pound, second converts to kilograms, and third is scale factor. Octal value is 2476_8, corresponding to about 0.90706E-5 seconds^2/pound.

$KM\underline{J}$ (KMJ_0, KMJ_1, KMJ_2): Single precision value of the one-jet acceleration about roll, pitch, and yaw axes respectively, scale factor B-13, units of revolutions/ (deci-second centi-second), so that multiplication of a time in centi-seconds by $KM\underline{J}$ yields a rotation rate in revolutions/deci-second. The quantities are computed in "S40.14", with pitch and yaw values the same.

KTLXdI: See Digital Autopilot TVC Routines.

LEMMASS: Single precision value of LM mass loaded by R2 of N47, and used in "MASSPROP" if bit 13 of DAPDATR1 = 0 to compute LM-attached mass properties constants. If bit 14 of DAPDATR1 = 1, is used in "FIXCW" to compute MASS (if DAPDATR1 set to indicate that LM not attached, cell contents ignored). Scale factor B16, units kilograms, but value should not exceed 2^{15} (or be equal to it) kilograms because of left shift in "MASSPROP": this is about 72000 pounds. For valid MASS computation, sum of CSMMASS and LEMMASS must be less than 2^{16} kilograms.

MASS: Single precision value of vehicle mass, program notation "WEIGHT/G", scale factor B16, units kilograms, loaded in "FIXCW" as the sum of CSMMASS and (if specified by DAPDATR1) LEMMASS. Note that the value in this cell is not updated by R03 ("DONOUN46") unless digit #1 of DAPDATR1 is 1, 2, 5, or 6. Cell is used as if it were double precision in "S40.1", "S40.1B", and "S40.13", and used as proper single precision quantity in "P37E". If used as "double precision" quantity, least significant half is AK_0.

MRKRTMP: See Burn Control.

NVWORD1: See Burn Control.

OPTIND: See Optics Computations.

PACTOFF: See Digital Autopilot TVC Routines.

PMANNDX: Single precision cell, scale factor B14, loaded in "STICKCHK" with bits 2-1 of TS (set before entering routine). These bits correspond to the "pitch" input, with 00_2 meaning no maneuver, 01_2 meaning a positive maneuver, 10_2 meaning a negative maneuver, and 11_2 treated the same as 00_2 except for rate-command use of the RHC (see "T5PHASE2"). TS is set to the "true" values of the input information (which for the controllers have an inversion from the binary values, so that a binary 0 channel input represents a logical 1).

RACFAIL: Single precision cell, scale factor B14, set in "S41.2" based on Quad failure information in DAPDATR2. It is set to 0 if Quads A and C are both indicated as functioning; it is set to +1 if Quad A is indicated as having failed; and it is set to -1 if Quad A is indicated as functioning and Quad C as not functioning. See information below.

RATEINDX: Single precision cell, scale factor B14, set in "S41.2" to bits 2-1 of DAPDATR1 (shifted left one place for convenience in indexing double precision constants). Value is 0 for a specified maneuver rate of 0.05°/second; value is 2 for a specified maneuver rate of 0.2°/second; value is 4 for a specified maneuver rate of 0.5°/second; and value is 6 for a specified maneuver rate of 2.0°/second.

RBDFAIL: Single precision cell, scale factor B14, set in "S41.2" based on Quad failure information in DAPDATR2. It is set to 0 if Quads B and D are both indicated as functioning; it is set to +1 if Quad B is indicated as having failed; and it is set to -1 if Quad B is indicated as functioning and Quad D as not functioning. See information below.

RCSFLAGS: Single precision flag word used for control of various RCS DAP functions. The individual bits have the following meanings:

Bit	Meaning
15	Set in "NEWANGL" on initial pass (also in R67) if RATEINDX = 6 (meaning large-rate R60 maneuver), and used in "T5PHASE2" to control setting of ATTKALMN (to -1 if bit is 1, and otherwise to 0) correspondingly. It avoids having attitude maneuver routine change ATTKALMN if RCS DAP filter is being started up. Reset e.g. in "STOPRATE".
14	Bit set in RCS DAP if rate filter must be initialized before computations are performed. It is set if e.g. bit 15 of channel 31 is 1(no G&N control), or if bit 6 of IMODES33 = 1, indicating IMU data not usable; it is reset e.g. in "REDAP".
13	Bit set 1 in "T5PHASE2" for a change in RHC inputs, to signify that roll damping must be completed before attitude"hold"resumed, and reset zero in "J23" if no roll firing required by phase-plane.
12	Bit set 1 in "T5PHASE2" for a change in RHC inputs, to signify that pitch damping must be completed before attitude"hold"resumed, and reset zero in "J23" if no pitch firing required by phase-plane.
11	Bit set 1 in "T5PHASE2" for a change in RHC inputs, to signify that yaw damping must be completed before attitude"hold"resumed, and reset zero in "J23" if no yaw firing required by phase-plane.
10-9	Bits set 0 in "T5PHASE2" and then set, for a change in RHC inputs, to the changed bits for roll(if any), to force a firing in "J23" (if not otherwise required by phase-plane) for that cycle, provided EDOT for that axis not too close to 0. Bit 1 if input 1→0 or 0→1.
8-7	Bits set 0 in "T5PHASE2" and then set, for a change in RHC inputs, to the changed bits for yaw (if any): see bits 10-9.
6-5	Bits set 0 in "T5PHASE2" and then set, for a change in RHC inputs, to the changed bits for pitch (if any): see bits 10-9.
4	Bit used in "KMATRIX" to cause "NEEDLER" to be performed on one pass (if presently 0) and a new <u>AK</u> value to be loaded on the next pass (a pass takes place once every 0.1 second).
3	Bit set to 1 to cause "NEEDLER" routine to perform the initialization functions for the FDAI display (reset error counter to 0 etc.). Reset by "NEEDLE11".
2	Bit set to 1 in "NEEDLE11" to signify that first pass of initialization function has been completed (causes, with bit 3 = 0, "NEEDLER2" to be entered at the next transfer to "NEEDLER"). Reset by "NEEDLER2".
1	Bit set 0 at end of "T6SETUP" and in "ZEROJET" to indicate to "T6START" that the next TIME6 interrupt that is valid should cause RWORD1, PWORD1, and YWORD1 to be loaded. The "T6START" routine then resets the bit to 1.

REPFRAC: See Digital Autopilot TVC Routines.

RMANNDX: Single precision cell, scale factor B14, loaded in "STICKCHK" with bits 6-5 of TS (set before entering routine), shifted right 4 places. These bits correspond to the "roll" input (see PMANNDX).

T5LOC: Address controlling where computations begin when program interrupt #2 (controlled by TIME5) is received. It is set to "T5IDLOC" to cause these program interrupts to be ignored: this setting is done as part of a "fresh start" and also in "STABLISH" if bits 14-13 of DAPDATR1 are 00_2. If a restart is encountered, the loading of T5LOC is controlled by bits 15-14 of FLAGWRD6, and T5LOC is set to "T5IDLOC", "REDORCS", "REDOTVC", and "REDOSAT" for bits 15-14 = 00_2, 01_2, 10_2, and 11_2 respectively.

For the Saturn DAP, T5LOC = "SATSTICK" as set by "REDOSAT" and "SATSTICK". "REDOSAT" is entered due to T5LOC setting to that address for a restart or by "SATSTKON" (entered from "STABLISH"). TIME5 setting is such as to cause "SATSTICK" to be entered every 0.1 second.

For the RCS DAP, startup is accomplished (from "STABLISH" or from "ENGINOFF", "V97E", or "V97T") by entering "RCSDAPON", which sets TIME5 to cause transfer to "RCSATT" (T5LOC setting) in 0.6 seconds. The delay if from "AUTO37" is 3.1 seconds. Two interrupts 20 ms apart cause entrance to "RCSATT", with subsequent branching controlled by T5PHASE (see Digital Autopilot RCS Routines). After the second "RCSATT" entrance, if filter not being initialized T5LOC is set to "JETSLECT" with another 20 ms delay; that routine in turn resets T5LOC to "RCSATT" with a 60 ms delay (for the 100 ms RCS DAP cycle rate).

For the TVC DAP, T5LOC is set to "T5IDLOC" in "IGNITION" shortly after engine-on, to disable RCS DAP entrances. About 0.4 seconds later, T5LOC is set to "TVCDAPON" (with delay of about 10 ms). "TVCDAPON" sets T5LOC to "TVCINIT1" (delay about 10 ms) which in turn, after entering "TVCINIT4", sets T5LOC to "DAPINIT" with delay from T5TVCDT (20 ms for LM-off). "DAPINIT" in turn sets T5LOC to "PITCHDAP" with delay of one complete TVC DAP cycle (40 ms for LM-off), causing entrance to "PITCHDAP" for LM-off about 0.48 seconds after engine-on. "PITCHDAP" sets T5LOC to "YAWDAP", which resets it to "PITCHDAP", each with delay from T5TVCDT, i.e. loop cycle rate of 40 ms for LM-off. The TVC DAP also uses waitlist task means to perform "TVCEXEC" every 0.5 second (starting .51 seconds after "TVCINIT4"), which in turn calls the roll DAP task, "ROLLDAP", 0.03 seconds after being entered. If a restart, "REDOTVC" sets T5LOC to "ENABL2", which in turn sets to "CMDSOUT", which in turn sets to either "TVCINIT1" or "DAPINIT" depending on timing of restart.

For the entry DAP, "CM/DAPON" sets T5LOC to "T5IDLOC", and the waitlist task "READGYMB", after entering "BODYRATE", sets it to "ATTRATES" (with 0.01 second delay). After executing the computations, T5LOC again set to "T5IDLOC" (at end of "CM/FDAIR"). The entry DAP cycle rate is defined by the 0.1-second waitlist calls of "READGYMB".

T6LOC: Address controlling where computations begin when program interrupt #1 (controlled by TIME6) is received. The interrupt is not generated unless TIME6 is counting down; contrary to TIME5, the countdown of TIME6 must be enabled by a channel bit (bit 15 of channel 13). It is set to "T5IDLOC" in "CM/DAPON" (since the Entry DAP does not make use of program interrupt #1).

In the RCS DAP, T6LOC is set to "T6START" in "ZEROJET" (which is also entered if no DAP or if Saturn DAP specified, to zero output channels). During normal RCS DAP operation, TIME6 is set to 14 ms in "JETSLECT", so that jets scheduled to be fired from about 86 ms to 100 ms will all be fired for the full 100 ms. Otherwise, TIME6 is set in "T6START" to the appropriate delay.

In the TVC DAP, T6LOC is used for timing of the roll RCS jets, and is set to "NOROLL1" in "JETROLL".

THETADX: See Digital Autopilot RCS Routines.

TIME5: Computer hardware clock register (cell 0030_8), incremented by 1 each 10 ms under hardware control. Its overflow (reaching a value of $2^{14} = 16384$) causes program interrupt #2 ("T5RUPT"), which is coded to start performing the computations whose starting address is in T5LOC. Setting TIME5 "to cause program interrupt #2 in 20 ms", for example, is accomplished by setting TIME5 = 37776_8.

TIME6: Computer hardware clock register (cell 0031_8), decremented by 1 each 0.625 ms (i.e. at a 1600 pps rate) under hardware control if bit 15 of channel 13 = 1. Because of decrement rate, can be considered to contain time-interval information with scale factor B10 in units of centi-seconds. Its reduction to zero causes program interrupt #1, which is coded to start performing the computations whose starting address is in T6LOC (via "T6RUPT").

TUSED: See Digital Autopilot Entry Routines.

TVCPHASE: See Digital Autopilot TVC Routines.

TVCPITCH: Single precision value of computer special erasable memory counter cell 0054_8 when used to provide pitch steering commands for the SPS engine gimbal (if bit 8 of channel 12 is 1). Pulses are provided from the cell to change the value of the associated error counter at a 3200 pps rate, until the cell contents reduced to zero, provided bit 11 of channel 14 is set. The computer's image of what is in the error counter is in PCMD (see Digital Autopilot TVC Routines), i.e. the "integral" of TVCPITCH output since error counter reset, by zeroing of bit 2 of channel 12, e.g. in "S40.6". One pulse gives 85.41 arc seconds of output, thus leading to the expression of scaling information as "B14 in units of CDU actuator output pulses" or B0 in units of ASCREV (actuator revolutions, namely 2^{14} pulses or about 1.07975111 revolution). There are about 42.14963 pulses per degree. The same cell is used for CDUSCMD (see Optics Computations), and hence optics use must be disabled when the cell is used for TVC purposes.

TVCYAW: Single precision value of computer special erasable memory counter cell 0053_8 when used to provide yaw steering commands for the SPS engine gimbal (if bit 8 of channel 12 is 1). Pulses are provided from the cell to change the value of the associated error counter at a 3200 pps rate, until the cell contents reduced to zero, provided bit 12 of channel 14 is set. The computer's image of what is in the error counter is in YCMD (see Digital Autopilot TVC Routines). Scaling same as TVCPITCH, and the same cell is also used for CDUTCMD (see Optics Computations).

VARK: See Digital Autopilot TVC Routines.

VARSTO: Single precision quantity computed in "MASSPROP" for use in "FIXCW" (cell not time-shared in TVC DAP), scale factor B20, units kilogram-meters^2, giving the value of IXX at "breakpoint" (when DELCMWT = 0).

VARST1: Single precision quantity computed in "MASSPROP", scale factor B20, used as the value of IAVG at "breakpoint", units kilogram-meters^2.

VARST2: Single precision quantity computed in "MASSPROP", scale factor B2, units seconds^2, used as the "breakpoint" value of IAVGdTLX.

VARST3: Single precision quantity computed in "MASSPROP", scale factor B-12, units seconds^2/kilogram, giving the slope of IAVGdTLX with CSMMASS for values above "breakpoint" (DELCMWT greater than 0).

VARST4: Single precision quantity computed in "MASSPROP", scale factor B6, units kilogram-meters^2 per kilogram, giving the slope of IAVG with CSMMASS for values above "breakpoint" (DELCMWT greater than 0).

VARST5: Single precision quantity computed in "MASSPROP", scale factor B6, units kilogram-meters^2 per kilogram, giving the slope of IXX with CSMMASS (both above and below "breakpoint"). Octal value is same for LM-off and LM-on (regardless of LEMMASS), including APS only.

VARST6: Single precision quantity computed in "MASSPROP", scale factor B6, units kilogram-meters^2 per kilogram, giving the slope of IAVG with CSMMASS for values not above "breakpoint" (DELCMWT not greater than 0).

VARST7: Single precision quantity computed in "MASSPROP", scale factor B-12, units seconds^2/kilogram, giving the slope of IAVGdTLX with CSMMASS for values not above "breakpoint" (DELCMWT not greater than 0).

VARST8: Single precision quantity computed in "MASSPROP", scale factor B20, units kilogram-meters^2, meaningful only for LM-on, and used to increment VARST1 if descent stage off: it is added to VARST1, but is computed for LM-on from two constants both of which are negative, hence a reduction in VARST1 results from the addition.

VARST9: Single precision quantity computed in "MASSPROP", scale factor B2, units seconds^2, meaningful only for LM-on, and used to increment VARST2 if descent stage off.

XTRANS: Single precision cell, scale factor B14, set 0 if bits 10 and 7 of DAPDATR1 are the same value (allows four-jet X-translation); set to +1 if bit 7 is 1 (allows use only of quad B/D for X-translation); and set to -1 if bit 10 is 1 (allows use only of quad A/C for X-translation). Setting performed in "S41.2". See information below.

YACTOFF: See Digital Autopilot TVC Routines.

YMANNDX: Single precision cell, scale factor B14, loaded in "STICKCHK" with bits 4-3 of TS (set before entering routine), shifted right 2 places. These bits correspond to the "yaw" input (see PMANNDX).

Summary of DAP Performance for RCS Failures

The only knowledge that the computer program has of RCS failures is by means of the information in DAPDATR2: there is no attempt made to deduce failures from observed spacecraft responses. The Entry DAP has no input provided to specify failed CM RCS jets, and the TVC roll DAP alternately uses (for rolls in a given direction) the A/C and B/D Quads, without regard to the RCS jet status information provided in DAPDATR2. Consequently, it is only the RCS DAP which makes use of jet status information furnished in DAPDATR2.

Routine "S41.2" serves as the interface between the information in DAPDATR2 and the RCS DAP jet selection logic (which is normally entered every 0.1 seconds while the RCS DAP is capable of providing output commands, and which starts at "JETSLECT"). Certain features of "S41.2" simplify the subsequent coding, but also have an influence upon the performance of the program that is not necessarily evidenced in the displays provided to the crew. These features include:

a) If Quad A is reported failed, the program assumes (without checking and with no special indication to the crew) that Quad C is functional.
b) If Quad B is reported failed, the program assumes (without checking and with no special indication to the crew) that Quad D is functional.
c) If neither Quad A/C or Quad B/D is specified for X-translation, the program assumes that both are to be used.
d) If only Quad A/C is specified for X-translation, no attempt is made to use Quad B/D for this purpose (even if Quad A or Quad C has a reported failure). Similarly for roll Quad selection.
e) If only Quad B/D is specified for X-translation, no attempt is made to use Quad A/C for this purpose (even if Quad B or Quad D has a reported failure). Similarly for roll Quad selection.

In addition to the features of "S41.2" that affect program performance, the jet selection logic itself has certain characteristics which strongly influence program performance in the presence of reported RCS failures. These include the following:

1) If only Quad A/C is specified for X-translation, and Quad A or Quad C has a reported failure, no X-translation is performed except that which may incidentally arise from the single-jet firings to perform pitch control (or yaw control in the event that Quad B or Quad D also has a reported failure).
2) Similar to #1, except with Quad B/D instead of Quad A/C.
3) If both Quad A/C and Quad B/D are specified for X-translation, proper translation from the unfailed Quad (plus that which may incidentally arise from single-jet firings) will be performed. The routine used to estimate burn time ("S40.13"), however, will perform its computations based on four-jet ullage. If Quads A/C and B/D both have reported failures, the situation of #1 applies.

4) If Quad A/C is specified for roll, and Quad A or Quad C has a reported failure, then no Y-translation is performed except that which may incidentally arise from the single-jet firings to perform roll control, "in which case the astronaut should switch to B/D roll."

5) If Quad B/D is specified for roll, and Quad B or Quad D has a reported failure, then no Z-translation is performed except that which may incidentally arise from the single-jet firings to perform roll control, "in which case the astronaut should switch to A/C roll."

6) If Quad A/C is specified for roll, and Quad B or Quad D has a reported failure, then single-jet Z-translation will be performed for that portion of the 0.1-second DAP cycle <u>not</u> required for roll attitude control (even though a roll torque is produced).

7) If Quad B/D is specified for roll, and Quad A or Quad C has a reported failure, then single-jet Y-translation will be performed for that portion of the 0.1-second DAP cycle <u>not</u> required for roll attitude control (even though a roll torque is produced).

8) If Quad A/C is specified for roll, and Quad B or Quad D has a reported failure, then the jet used in #6 will also be fired during the portion of the 0.1-second DAP cycle that <u>is</u> used for roll attitude control if and only if the roll torque on the vehicle is not reduced to 0 during this interval. It would be fired if no Quad A/C failures were reported (since two-jet torque is used) and no Y-translation is simultaneously specified; other cases can best be analyzed by review of the jet table information itself (see Digital Autopilot RCS Routines).

9) If Quad B/D is specified for roll, and Quad A or Quad C has a reported failure, then the jet used in #7 will also be fired during the portion of the 0.1-second DAP cycle that <u>is</u> used for roll attitude control if and only if the roll torque on the vehicle is not reduced to 0 during this interval. It would be fired if no Quad B/D failures were reported and no Z-translation is simultaneously specified; see jet table information for other conditions.

10) As suggested by #1 - #3 above, the philosophy of #6 - #9 above for Y-translation and Z-translation is <u>not</u> employed for X-translation.

Digital Autopilot RCS Routines

RCSDAPON

Set TIME5 to cause program interrupt #2 in 0.6 seconds

T5PHASE = 16324 (i.e. positive non-zero)

Set bit 3 of RCSFLAGS = 1 (cause FDAI error display initialization)

T5LOC = "RCSATT"

Set bits 15-14 (DAPBIT1, DAPBIT2) of FLAGWRD6 = 01_2

Return

REDORCS

If T5PHASE $\leqslant$ 0:

  T5PHASE = -1

T5LOC = "RCSATT"

Proceed to "RCSATT"

RCSATT

TS = channel 31

If bits 14-13 of C31FLWRD $\neq 00_2$:

  TS = C31FLWRD

If bit 15 (G&N Autopilot Control complement) of TS = 1:

  Set bit 14 of RCSFLAGS = 1

  HOLDFLAG = +MAX

  $\underline{\text{ERROR}}$ = +0

  If bit 14(Free Mode complement) of TS = 1: (TS re-computed)

    T5PHASE = 8192 (i.e. positive non-zero)

    Set TIME5 to cause program interrupt #2 in 0.1 second

    Perform "ZEROJET"

    Proceed to "KMATRIX"

If T5PHASE > 0: (Tag here "SETT5")

  HOLDFLAG = 1

  Proceed to "REDAP"

If T5PHASE = +0, proceed to "T5PHASE2"

If T5PHASE $<$ 0, proceed to "REDAP"

T5PHASE = +0 (was -0, indicating phase 1)

T5TIME = TIME5

Set TIME5 to cause program interrupt #2 in 0.02 seconds

If bit 6(NOIMUDAP) of IMODES33 = 1: (IMU data not usable)

  Set bit 14 of RCSFLAGS = 1

  HOLDFLAG = 8192

  TS = channel 31

  If bits 14-13 of C31FLWRD $\neq 00_2$:

    TS = C31FLWRD

  If bit 14(Free Mode complement) of TS = 0, Resume

  Set TIME5 to cause program interrupt #2 in 0.2 seconds

  T5PHASE = 16364 (i.e. positive non-zero)

  Resume

If bit 14 of RCSFLAGS = 1, proceed to "KMATRIX" (Tag here "RATEFILT")

$DRH\underline{O} = DRH\underline{O} - K_{gnl_{ATTKALMN}} \; DRH\underline{O}_{sp} - ADO\underline{T}$

Set $RHO_0 = CDU_x$ and $TS = RHO_0$

$TS_1 = -(TS - RHO_0)$ (ones complement difference formed)

Set $RHO_1 = CDU_y$ and $TS = RHO_1$

$TS_2 = -(TS - RHO_1)$ (ones complement difference formed)

$DELTEMP_x = TS_1 + AMGB1 \; TS_2$

Set $RHO_2 = CDU_z$ and $TS = RHO_2$

$TS_3 = -(TS - RHO_2)$ (ones complement difference formed)

$DELTEMP_y = AMGB4 \; TS_2 + AMGB5 \; TS_3$

$DELTEMP_z = AMGB7 \; TS_2 + AMGB8 \; TS_3$

$DRH\underline{O} = DRH\underline{O} + DELTEM\underline{P}$

$MERRO\underline{R} = MERRO\underline{R} + DELTEM\underline{P}$

$\underline{ADOT} = \underline{ADOT} + K_{gn2_{ATTKALMN}} \underline{DRHO}_{sp} + \underline{KMJ}\ \underline{DFT}$

(last term involves e.g. KMJ_1 DFT_1 for $ADOT_1$)

Proceed to "KMATRIX"

KMATRIX

TS = bits 4-1 of ATTSEC

If TS = 0:

Establish "AMBGUPDT" (priority 34_8)

ATTSEC = 9

If TS $\neq$ 0:

ATTSEC = TS - 1

If HOLDFLAG < 0:

$\underline{CDUXD} = \underline{CDUXD} + \underline{DELCDU}$, in range $\pm$ 180°

$THETAD\underline{X} = \underline{CDUXD}_{sp}$

If bit 4 of RCSFLAGS = 0:

Set bit 4 of RCSFLAGS = 1

Perform "NEEDLER"

Resume

Set bit 4 of RCSFLAGS = 0

If bit 9(NEEDLFLG) of FLAGWRD0 = 0: (means do not display total attitude error; V62 and V63 set 1; V61 sets 0)

$\underline{AK} = - \underline{ERROR}$

Resume

If bit 6(N22ERNDS) of FLAGWRD9 = 1: (set 1 by V62, 0 by V63)

$\underline{ANGREF} = \underline{THETAD}$ ($\underline{THETAD}$ is N22)

If bit 6(N22ERNDS) of FLAGWRD9 = 0:

$\underline{ANGREF} = \underline{CPHIX}$ ($\underline{CPHIX}$ is N17)

$\underline{TS} = \underline{CDU}$

$AK_0 = (ANGREF_x - TS_x) + AMGB1\ (ANGREF_y - TS_y)$

$AK_1 = AMGB4\ (ANGREF_y - TS_y) + AMGB5\ (ANGREF_z - TS_z)$

$AK_2 = AMGB7\ (ANGREF_y - TS_y) + AMGB8\ (ANGREF_z - TS_z)$

Resume

<u>AMBGUPDT</u> (Established once a second by "KMATRIX", and also by "REDAP").

If bits 15-14 (DAPBIT1, DAPBIT2) of FLAGWRD6 $\neq 01_2$, End of job

$AMGB1 = \sin_{sp} CDU_z$

$AMGB8 = \cos_{sp} (CDU_x - K_{quadan})$ (ones complement difference formed when K_{quadan} used)

$AMGB4 = \cos_{sp}(CDU_x - K_{quadan}) \cos_{sp} CDU_z$

$AMGB5 = \sin_{sp} (CDU_x - K_{quadan})$

$AMGB7 = - \sin_{sp}(CDU_x - K_{quadan}) \cos_{sp} CDU_z$

End of job

<u>REDAP</u>

Perform "S41.2"

Perform "S40.14"

$\underline{ERROR} = 0$

$\underline{BIAS} = 0$

$\underline{TAU} = 0$

ATTSEC = 0

$\underline{DRHO} = 0$

$\underline{DFT} = 0$

$\underline{MERROR} = 0$

$\underline{ADOT} = 0$

$\underline{WBODY} = 0$

Perform "ZEROJET"

$CHANTEMP = 77777_8$

CH31TEMP = 77777_8

SLOPE = K_{p24}

T5TIME = 4

ATTKALMN = 11

RH$\underline{O}$ = CD$\underline{U}$

T5PHASE = +0

If bit 6(NOIMUDAP) of IMODES33 = 1: (IMU data not usable)

    ATTKALMN = 0

    RCSFLAGS = 20004_8 (bits 14 and 3 = 1)

If bit 6(NOIMUDAP) of IMODES33 = 0:

    Establish "AMBGUPDT" (priority 34_8)

    RCSFLAGS = 00004_8 (bit 3 = 1)

Set TIME5 to cause program interrupt #2 in 0.06 seconds

Resume

<u>ZEROJET</u>

BLASTi+1 = 0 (i = 0,1,2)

BLASTi+0 = 0 (i = 0,1,2)

iWORD2 = 0 (i = R,P,Y)

iWORD1 = 0 (i = R,P,Y)

BLAST1+1 = 4

BLAST2+1 = 11

Set bit 1 of RCSFLAGS = 0

T6LOC = "T6START"

Set TIME6 to 23 (cause program interrupt #1 in about 14 ms)

Perform "C13STALL"

Set bit 15(TIME6 Count Enable) of channel 13 = 1

Return

T5PHASE2

If ATTKALMN $>$ 0:

ATTKALMN = ATTKALMN - 1

TS = 8 - T5TIME

If TS $\leq$ 0:

TS = 2

Set TIME5 to cause program interrupt #2 in TS centi-seconds

T5PHASE = -0

Resume

Set TS = TIME5 and TIME5 to cause program interrupt #2 in 0.02 seconds

T5TIME = T5TIME + TS

If$(|CDU_z| + K_{m75degs}) > 0$:

Set bit 14(STIKFLAG) of FLAGWRD1 = 1

HOLDFLAG = +0

If bit 15 of RCSFLAGS = 1: (RATEINDX = 6 in "NEWANGL" for R60/R67)

ATTKALMN = -1

If bit 15 of RCSFLAGS = 0:

ATTKALMN = 0

Set bits 10-5 of RCSFLAGS = 0 (initialize forced firing bits)

TS = channel 31 (bits 6-1 are RHC command complements)

TS_1 = bits 6-1 of$\left(CH31TEMP \cap (-TS) + (-CH31TEMP) \cap TS \right)$

(gives ones in bits whose TS values differ from CH31TEMP value)

If $TS_1 \neq 0$:

CH31TEMP = TS

RCSFLAGS = RCSFLAGS + 2^4 TS_1 (puts TS_1 into bits 10-5)

Set bits 13-11 of RCSFLAGS = 1 (rate damping)

TS = bits 6-1 of (- CH31TEMP) (gives 1 where logic ones, i.e. binary zeros, were in input)

If TS = 0, proceed to "AHFNOROT" (no RHC inputs)

HOLDFLAG = TS

Perform "STICKCHK"

Set bit 14(STIKFLAG) of FLAGWRD1 = 1

TS = channel 31

If bits 14-13 of C31FLWRD $\neq 00_2$:

    TS = C31FLWRD

If bit 14(Free Mode complement) of TS = 0:

    TS_2 = (RCSFLAGS shifted right 4 places) (changed bits back to bits 6-1)

    TS = bits 6-1 of $\left(TS_2 \cap (-\ CH31TEMP) \right)$

    Proceed to 9th line of "AHFNOROT" (TS bits 1 for 1 → 0 transition)

If bit 14 of RCSFLAGS = 1:

    Set TIME5 to cause program interrupt #2 in 0.2 second

    T5PHASE = 16364 (i.e. positive non-zero)

    Resume

If RATEINDX - 5 > 0:

    ATTKALMN = -1

Perform the following for i = 2,1,0 (j = Y, P, R respectively):

    If jMANNDX = 0:

        $WBODY_i = 0$

    If jMANNDX $\neq$ 0:

        TS = jMANNDX + RATEINDX - 1

        $WBODY_i = K_{mantab_{TS}}$

        If bits 13-11 of RCSFLAGS = 0: (damping done)

            $MERROR_i = MERROR_i - K_{mantab_{TS}}$

    If bits 13-11 of RCSFLAGS $\neq$ 0:

        $MERROR_i = 0$

    $ERROR_i = MERROR_{i_{sp}}$

TS_1 = ADB + K_{afrg}

i = 2

Proceed to "JLOOP"

AHFNOROT

TS = channel 31

If bits 14-13 of C31FLWRD $\neq$ 00_2:

TS = C31FLWRD

If bit 14(Free Mode complement) of TS = 0:

HOLDFLAG = 1 (Tag here "FREECONT")

TS_1 = channel 32 (bits 6-1 are minimum impulse command complements)

TS = bits 6-1 of$\left(\text{CHANTEMP} \cap (-\ TS_1)\right)$ (gives ones in bits changed from 1 to 0)

CHANTEMP = TS_1

Perform "STICKCHK" (Enter here from "T5PHASE2" for RHC minimum impulse; tag "RHCMINP")

$TAU_0 = K_{mintu_{RMANNDX}}$

$TAU_1 = K_{mintu_{PMANNDX}}$

$TAU_2 = K_{mintu_{YMANNDX}}$

$ERROR\underline{R}$ = 0

T5LOC = "JETSLECT"

Resume

If bit 14 of RCSFLAGS = 1:

Set TIME5 to cause program interrupt #2 in 0.2 seconds

T5PHASE = 16364 (i.e. positive non-zero)

Resume

If bit 13(Hold Mode complement) of TS = 0: (TS re-computed)

If HOLDFLAG $\neq$ +0: (Tag here "HOLDFUNC")

$WBOD\underline{Y}$ = 0

$BIA\underline{S}$ = 0

(If HOLDFLAG $\neq$ +0):

If bits 13-11 of RCSFLAGS $\neq$ 0:

$\underline{ERROR} = 0$

$TS_1 = ADB + K_{afrg}$

$i = 2$

Proceed to "JLOOP"

HOLDFLAG = +0

$\underline{THETADX} = \underline{CDU}$

If bit 13(Hold Mode complement) of TS = 1: (since bit 14 = 1, this means Auto)

If HOLDFLAG > 0:

$\underline{WBODY} = 0$

$\underline{BIAS} = 0$

If bits 13-11 of RCSFLAGS $\neq$ 0:

$\underline{ERROR} = 0$

$TS_1 = ADB + K_{afrg}$

$i = 2$

Proceed to "JLOOP"

HOLDFLAG = +0

$\underline{THETADX} = \underline{CDU}$

$TS_1 = CDU_y - THETADY$ (ones complement difference formed) (Tag "ATTHOLD")

$TS_2 = CDU_z - THETADZ$ (ones complement difference formed)

$ERROR_0 = (CDU_x - THETADX) + AMGB1\ TS_1$ (ones complement diff.)

$ERROR_1 = AMGB4\ TS_1 + AMGB5\ TS_2$

$ERROR_2 = AMGB7\ TS_1 + AMGB8\ TS_2$

If HOLDFLAG $<$ 0:

$\underline{ERROR} = \underline{ERROR} + \underline{BIAS}$

$TS_1 = ADB + K_{afrg}$

$i = 2$

Proceed to "JLOOP"

JLOOP

$EDOT = ADOT_i$

If HOLDFLAG $\neq$ 0:

$EDOT = EDOT - WBODY_i$

$AERR = ERROR_i$

$ADBVEL = TS_1 \operatorname{sgn} EDOT$ (-0 is negative)

$AERRVEL = AERR \operatorname{sgn} EDOT$ (-0 is negative)

If $AERRVEL - ADB + K_{whdsop} \leq 0$: (tag "J6.")

If $(|EDOT| - K_{wimh}) > 0$: (tag "J8")

$TS = K_{wim} \operatorname{sgn} EDOT - EDOT$ (tag "J22")

Proceed to "JTIME"

If $|EDOT| / SLOPE + TS_1 + AERRVEL \geq 0$: (tag "NJ22")

Proceed to "J23"

If $|EDOT| - K_{wimmh} > 0$, proceed to "J23"

If $AERRVEL + TS_1 + K_{wmhdsop} \leq 0$: (tag "NJ23")

$TS = K_{wim} \operatorname{sgn} EDOT - EDOT$ (tag "J22")

Proceed to "JTIME"

$TS = - EDOT - K_{sope2} (AERR + ADBVEL)$ (tag "J24")

Proceed to "JTIME"

If AERRVEL − $TS_1 \leq 0$:

If ADB − AERRVEL − |EDOT| / SLOPE ≤ 0:

TS = − EDOT (tag "J18")

Proceed to "JTIME"

Proceed to "J23"

If AERRVEL − TS_1 − $K_{wmhdsop} \leq 0$ (tag "J7")

TS = K_{sope2} (ADBVEL − AERR) − EDOT (tag "J20")

Proceed to "JTIME"

TS = − K_{wim} sgn EDOT − EDOT (tag "J21")

Proceed to "JTIME"

<u>J23</u> Entered if conclude in deadzone.

j = 13 − i

Set bit j of RCSFLAGS = 0 (reset rate damping bit)

If i = 2:

TS = 00300_8 (bits 8-7, yaw command)

If i = 1:

TS = 00060_8 (bits 6-5, pitch command)

If i = 0:

TS = 01400_8 (bits 10-9, roll command)

If RCSFLAGS $\cap$ TS = 0: (no forced firing from "T5PHASE2" RHC)

Proceed to second line of "JTIME" (note that TAU_i <u>not</u> explicitly set 0)

TS = − EDOT (tag "J18")

Proceed to "JTIME"

<u>JTIME</u>

TAU_i = JdM_i TS, with magnitude limited < 2^{10} centi-seconds

If $i > 0$:

$i = i - 1$

Proceed to "JLOOP"

T5LOC = "JETSLECT"

Resume

<u>JETSLECT</u>

TS = 6 - T5TIME

If TS $\leq$ 0:

TS = 2

Set TIME5 to cause program interrupt #2 in TS centi-seconds

TIME6 = K_{14ms} (note that any jets desired to be on for interval $\sim$86-100 ms will remain on for full 100 ms)

Perform "C13STALL"

Set bit 15 (TIME6 Count Enable) of channel 13 = 1

TS = bits 12-7 of - (channel 31) (gives "true" values of translation hand controller bits)

If TS = 0:

XNDX1 = 0

XNDX2 = 0

YNDX = 0

ZNDX = 0

If TS $\neq$ 0:

XNDX1 = bits 8-7 of TS, shifted right 6 places (to bits 2-1)

XNDX2 = XNDX1

YNDX = bits 10-9 of TS, shifted right 8 places (to bits 2-1)

ZNDX = bits 12-11 of TS, shifted right 10 places (to bits 2-1)

(If TS $\neq$ 0):

If bit 14 of DAPDATR1 = 0: (means LM-off)

ATTKALMN = -2

If bit 14 of DAPDATR1 = 1: (means LM-on)

ATTKALMN = -3

If XTRANS = 1:

XNDX1 = 0

If XTRANS = -1:

XNDX2 = 0

If TAU_1 = 0:

PINDEX = 0

If $TAU_1 > 0$:

PINDEX = 1

If $TAU_1 < 0$:

PINDEX = 2

If RACFAIL = 0:

$TS = K_{xtndx_{XNDX1}}$

If RACFAIL > 0:

TS = 9

If RACFAIL < 0:

TS = 12

TS = TS + PINDEX

PWORD1 = bits 10, 9, and 4-1 of $K_{pytab_{TS}}$

NPJETS = bits 10-9 of PWORD1, shifted right 8 places (to bits 2-1)

If $TAU_2 = 0$:

YINDEX = 0

If $TAU_2 > 0$:

YINDEX = 1

If $TAU_2 < 0$:

YINDEX = 2

If RBDFAIL = 0:

$TS = K_{xtndx_{XNDX2}}$

If RBDFAIL > 0:

TS = 9

If RBDFAIL < 0:

TS = 12

TS = TS + YINDEX

YWORD1 = bits 12, 11, and 8-5 of $K_{pytab_{TS}}$

NYJETS = bits 12-11 of YWORD1, shifted right 10 places (to bits 2-1)

If $TAU_0 = 0$:

RINDEX = 0

If $TAU_0 > 0$:

RINDEX = 1

If $TAU_0 < 0$:

RINDEX = 2

If ACORBD $\geq$ +0, proceed to "BDROLL"

If RACFAIL = 0:

$TS = K_{xtndx_{YNDX}}$

If RACFAIL > 0:

TS = 9

If RACFAIL < 0:

TS = 12

TS = TS + RINDEX

RWORD1 = bits 11-5 of $K_{rtab_{TS}}$

If ZNDX ≮ 0:

NRJETS = (bits 11-9 of RWORD1, shifted right 8 places) - 2

Proceed to "ROLLTIME"

If RBDFAIL = 0:

TS = 0

If RBDFAIL > 0:

TS = 3

If RBDFAIL < 0:

TS = 6

$TS = TS + K_{xnlndx_{ZNDX}}$

TS_1 = RWORD1 + (bits 11-9, 4-1 of $K_{yztab_{TS}}$)

NRJETS = (bits 11-9 of TS_1, shifted right 8 places) - 4

If NRJETS ≠ 0: (have a net roll torque)

RWORD1 = TS_1

Proceed to "ROLLTIME"

If $TAU_0 = 0$: (i.e. no roll requested)

$RWORD1 = TS_1$

Proceed to "ROLLTIME"

NRJETS = (bits 11-9 of RWORD1, shifted right 8 places) - 2

Proceed to "ROLLTIME" (Z-translation ignored in favor of roll)

<u>BDROLL</u>

If RBDFAIL = 0:

$TS = K_{xtndx_{ZNDX}}$

If RBDFAIL > 0:

TS = 9

If RBDFAIL < 0:

TS = 12

TS = TS + RINDEX

RWORD1 = bits 14-12 and 4-1 of $K_{rtab_{TS}}$

If YNDX = 0:

NRJETS = (bits 14-12 of RWORD1, shifted right 11 places) - 2

Proceed to "ROLLTIME"

If RACFAIL = 0:

TS = 0

If RACFAIL > 0:

TS = 3

If RACFAIL < 0:

TS = 6

$TS = TS + K_{xnlndx_{YNDX}}$

$TS_1 = RWORD1 + (\text{bits 14-12, 8-5 of } K_{yztab_{TS}})$

NRJETS = (bits 14-12 of TS_1, shifted right 11 places) - 4

If NRJETS $\neq$ 0: (have a net roll torque)

RWORD1 = TS_1

Proceed to "ROLLTIME"

If TAU_0 = 0: (no roll requested)

RWORD1 = TS_1

Proceed to "ROLLTIME"

NRJETS = (bits 14-12 of RWORD1, shifted right 11 places) - 2

Proceed to "ROLLTIME" (Y-translation ignored in favor of roll)

<u>ROLLTIME</u>

If TAU_0 = 0:

$DFT_0 = K_{dftm_{NRJETS}}$

RWORD2 = RWORD1

BLASTO+0 = 0

Proceed to "PITCHTIM"

BLASTO+0 = $K_{njt_{NRJETS}}$ TAU_0

TS = BLASTO+0 + K_{mptlsc}

If TS > 0:

$DFT_0 = K_{dftm_{NRJETS}}$

$TAU_0 = TAU_0 - DFT_0$

BLASTO+0 = K_{ptlsc}

If TS $\leq$ 0:

Limit BLASTO+0 $\geq K_{14ms}$

DFT_0 = NRJETS BLASTO+0

TAU_0 = 0

If YNDX = 0:

RWORD2 = 0

If ZNDX = 0, proceed to "PITCHTIM"

Proceed to "ACBD2Z"

If ACORBD $\not<$ -0:

If $|$RACFAIL$|$ > 0:

RWORD2 = 0

If ZNDX = 0, proceed to "PITCHTIM"

Proceed to "ACBD2Z"

TS = $K_{xtndx_{YNDX}}$

RWORD2 = bits 11-5 of $K_{rtab_{TS}}$

If ZNDX = 0, proceed to "PITCHTIM"

Proceed to "ACBD2Z"

If RACFAIL = 0:

TS = 0

If RACFAIL > 0:

TS = 3

If RACFAIL < 0:

TS = 6

TS = TS + $K_{xnlndx_{YNDX}}$

RWORD2 = bits 14-12, 8-5 of $K_{yztab_{TS}}$

NRJETS = (bits 14-12 of RWORD2, shifted right 11 places) - 2

$DFT_0 = DFT_0 + NRJETS\ (K_{ptlsc} - BLASTO+O)$

If ZNDX = 0, proceed to "PITCHTIM"

Proceed to "ACBD2Z"

ACBD2Z

If ACORBD $\geq$ +0:

If $|RBDFAIL| > 0$, proceed to "PITCHTIM"

$TS = K_{xtndx_{ZNDX}}$

$RWORD2 = RWORD2 + (\text{bits 14-12, 4-1 of } K_{rtab_{TS}})$

Proceed to "PITCHTIM"

If RBDFAIL = 0:

TS = 0

If RBDFAIL > 0:

TS = 3

If RBDFAIL < 0:

TS = 6

$TS = TS + K_{xnlndx_{ZNDX}}$

$RWORD2 = RWORD2 + (\text{bits 11-9, 4-1 of } K_{yztab_{TS}})$

NRJETS = (bits 14-9 of RWORD2, shifted right 8 places) - 2

$DFT_0 = DFT_0 + NRJETS\ (K_{ptlsc} - BLASTO{+}0)$

Proceed to "PITCHTIM"

PITCHTIM

If $TAU_1 = 0$:

$DFT_1 = 0$

PWORD2 = PWORD1

BLAST1+0 = 0

Proceed to "YAWTIME"

If $TAU_1 < 0$:

NPJETS = - NPJETS

$BLAST1+0 = K_{njt_{NPJETS}} \quad TAU_1$

$TS = BLAST1+0 + K_{mptlsc}$

If $TS > 0$:

$DFT_1 = K_{dftm_{NPJETS}}$

$TAU_1 = TAU_1 - DFT_1$

$BLAST1+0 = K_{ptlsc}$

If $TS < 0$:

Limit $BLAST1+0 > K_{14ms}$

$DFT_1 = NPJETS \quad BLAST1+0$

$TAU_1 = 0$

$TS = 0$

If $RACFAIL = 0$:

$TS = K_{xtndx_{XNDX1}}$

PWORD2 = bits 10, 9, and 4-1 of $K_{pytab_{TS}}$ (is 0 for TS = 0)

Proceed to "YAWTIME".

YAWTIME

If $TAU_2 = 0$:

$DFT_2 = 0$

YWORD2 = YWORD1

$BLAST2+0 = 0$

Proceed to "T6SETUP"

If $TAU_2 < 0$:

$NYJETS = - NYJETS$

$BLAST2+0 = K_{njt_{NYJETS}} \quad TAU_2$

$TS = BLAST2+0 + K_{mptlsc}$

If TS > 0:

$DFT_2 = K_{dftm_{NYJETS}}$

$TAU_2 = TAU_2 - DFT_2$

$BLAST2+0 = K_{ptlsc}$

If TS ≤ 0:

Limit BLAST2+0 ≥ K_{14ms}

DFT_2 = NYJETS BLAST2+0

$TAU_2 = 0$

TS = 0

If RBDFAIL = 0:

$TS = K_{xtndx_{XNDX2}}$

YWORD2 = bits 12, 11, and 8-5 of $K_{pytab_{TS}}$ (is 0 for TS = 0)

Proceed to "T6SETUP"

T6SETUP

BLAST0+1 = 0

BLAST1+1 = 4

BLAST2+1 = 11

If BLAST1+0 - BLAST0+0 ≤ 0:

Set BLAST0 = BLAST1 and BLAST1 = BLAST0

If BLAST2+0 - BLAST1+0 ≤ 0:

Set BLAST1 = BLAST2 and BLAST2 = BLAST1

If BLAST1+0 - BLAST0+0 ≤ 0:

Set BLAST0 = BLAST1 and BLAST1 = BLAST0

BLAST2+0 = BLAST2+0 - BLAST1+0 (now ordered BLAST0+0 ≤ BLAST1+0 ≤ BLAST2+0)

BLAST1+0 = BLAST1+0 - BLAST0+0

T5LOC = "RCSATT"

Set bit 1 of RCSFLAGS = 0

T5PHASE = -0

Resume

<u>T6START</u>

If $|TIME6| > 0$, Resume

If bit 1 of RCSFLAGS = 0:

Set bit 1 of RCSFLAGS = 1

Channel 6 = bits 8-1 of RWORD1

Channel 5 = bits 8-1 of (PWORD1 + YWORD1)

If BLAST0+0 > 0:

TIME6 = BLAST0+0

BLAST0+0 = 0

Perform "C13STALL"

Set bit 15(TIME6 count enable) of channel 13 = 1

Resume

If BLAST0+0 = 0:

TS = BLAST0+1

Perform "REPLACE"

BLAST0+0 = - 1/16

If BLAST1+0 > 0:

TIME6 = BLAST1+0

BLAST1+0 = 0

Perform "C13STALL"

Set bit 15(TIME6 count enable) of channel 13 = 1

Resume

If BLAST1+0 = 0:

TS = BLAST1+1

Perform "REPLACE"

BLAST1+0 = -1/16

If BLAST2+0 > 0:

TIME6 = BLAST2+0

BLAST2+0 = 0

Perform "C13STALL"

Set bit 15 (TIME6 count enable) of channel 13 = 1

Resume

If BLAST2+0 = 0:

TS = BLAST2+1

Perform "REPLACE"

BLAST2+0 = -1/16

Resume

<u>REPLACE</u>

If TS = 0: (roll information)

Channel 6 = bits 8-1 of RWORD2

Return

If TS = 4: (pitch information)

TS_1 = bits 8-5 of channel 5

Channel 5 = bits 8-1 of (TS_1 + PWORD2)

Return

If TS = 11: (yaw information)

TS_1 = bits 4-1 of channel 5

Channel 5 = bits 8-1 of (TS_1 + YWORD2)

Return

Quantities in Computations

See also list of major variables and list of routines

ACORBD: See Digital Autopilot Interface Routines.

ADB: See Digital Autopilot Interface Routines.

ADBVEL: Single precision value of (ADB + K_{afrg}) sgn EDOT used in "JLOOP", scale factor B-1, units revolutions.

ADO$\underline{T}$ ($ADOT_0$, $ADOT_1$, $ADOT_2$): Filtered value of roll, pitch, and yaw body-angle rates (about "control axes"), scale factor B-3, units revolutions/deci-second (one deci-second is 0.1 seconds, the iteration rate of RCS DAP). Alternatively, could be considered scaled B0 in "units" of 450°/second. Cells are OMEGA$\underline{B}$ in TVC DAP.

AERR: Single precision value of $ERROR_1$ about appropriate axis used in "JLOOP", scale factor B-1, units revolutions.

AERRVEL: Single precision value of AERR sgn EDOT used in "JLOOP", scale factor B-1, units revolutions.

A$\underline{K}$ (AK_0, AK_1, AK_2): See Digital Autopilot Interface Routines.

AMGB1, AMGB4, AMGB5, AMGB7, AMGB8: Single precision matrix elements which are used by RCS DAP to transform information from gimbal to control axes, scale factor B0, computed once per second by "AMBGUPDT". The matrix from gimbal to control axes may be written as:

$$\begin{bmatrix} 1 & AMGB1 & 0 \\ 0 & AMGB4 & AMGB5 \\ 0 & AMGB7 & AMGB8 \end{bmatrix}$$

In order to "account for the roll displacements of the reaction jets with respect to navigation base coordinates", the outer gimbal angle (CDU_x) used to compute the matrix elements is decremented by K_{quadan}, thus giving the "control axes" result.

ANGRE$\underline{F}$: Value of reference angle (THETA$\underline{D}$ or CPHI$\underline{X}$) used in "KMATRIX" to compute display information of total attitude error (if required), scale factor B-1, units revolutions.

ATTKALMN: Single precision quantity, scale factor B14, used to select appropriate gains in "RCSATT". To initialize the filter, cell is set to 11 (decremented to 10 before used) in "REDAP", and decremented in "T5PHASE2" (with subsequent performance of further computations, including jet firings, bypassed until ATTKALMN = 0). A value of 0

is the normal steady-state filter gain value; a setting of -1 is used for high-rate (RATEINDX = 6) automatic maneuvers (set in "T5PHASE2" if bit 15 of RCSFLAGS = 1) or RHC-commanded maneuvers at high rate; a setting of -2 is used (set in "JETSLECT") if translation hand controller inputs with LM-off are provided; and a setting of -3 is used for translation inputs with LM-on. The reset to 0 is accomplished (from negative values) in "T5PHASE2" if bit 15 of RCSFLAGS = 0.

ATTSEC: Single precision quantity, scale factor B14, initialized to 0 in "REDAP" and used in "KMATRIX" to cause "AMBGUPDT" to be established once a second (every tenth entrance to "KMATRIX").

BIAS ($BIAS_0$, $BIAS_1$, $BIAS_2$): Set of single precision angles, scale factor B-1, units revolutions, "added to the attitude errors to provide additional lead and prevent overshoot when starting an automatic maneuver." They are added to ERROR (used as the attitude error in "JLOOP") if HOLDFLAG negative, and the values are computed in "LOCSKIRT" and "MANUSTOP" (where set 0). Magnitude corresponds to the value of (1/0.6) times maneuver rate (see K_{biasc} in Attitude Maneuvers), and otherwise is 0.

BLAST0, BLAST1, BLAST2: Set of 3 quantities handled double precision, used for RCS jet burn time control. Before "T6SETUP" is entered, BLAST0 is associated with roll, BLAST1 with pitch, and BLAST2 with yaw. The most significant halves of each word have required firing time for RWORD1, PWORD1, and YWORD1 respectively, scale factor B10, units centi-seconds. The least significant halves contain an index which identifies the axis in the most significant half (set at start of "T6SETUP" to 0, 4, and 11 respectively for convenience in the coding). Cells subsequently arranged in order of increasing time durations, then replaced with time intervals. In "T6START", most significant half set +0 if associated time in TIME6, and -1 (B14, i.e. -1/16 B10) after the expiration of that interval.

C31FLWRD: Single precision flagword used for backup of bits 15-13 of channel 31 (by bits 15-13, i.e. digit "A") and bits 5-4 of channel 33 (by bits 5-4, i.e. digit "D"). It can be loaded by N01 means (address 0374_8). For digit A, a value of 0 (or 4) means should use bits 15-13 of channel 31; values of 1-3 mean G&N Control (Free, Attitude Hold, and Auto respectively); values of 5-7 mean SCS Control. For digit D, a value of 0 means use bits 5-4 of channel 33; of 1-3 mean optics CMC, Zero, and Manual respectively (values of 4-7 mean the same as 0-3 respectively). Should be pad-loaded to zero; digits B, C, and E ("ABCDE") are ignored.

CDUXD: See Attitude Maneuvers (used in "KMATRIX" to update THETADX if HOLDFLAG is negative).

CH31TEMP: Single precision cell used in "T5PHASE2" to contain the previous value of channel 31 inputs (bits 6-1 are rotational hand controller inputs) if bits 6-1 change, in order to detect binary 1 → 0 transitions (logic 0 to 1) for RHC control purposes: binary 0 → 1 are also detected, of course. It is initialized to 77777_8 by "REDAP".

CHANTEMP: Single precision cell used in Free Mode (at the start of "AHFNOROT") to retain previous input of channel 32 (cf. CH31TEMP), for use in providing "minimum impulse" control. Initialized to 77777_8 in "REDAP".

CPHI$\underline{X}$: Set of reference angles for attitude error (A$\underline{K}$) display, scale factor B-1, units revolutions. They can be loaded by N17, or set to present CD$\underline{U}$ values by verb 60 (and displayed on FDAI by verb 63).

DELCD$\underline{U}$: See Attitude Maneuvers (used in "KMATRIX" to increment CDUX$\underline{D}$ values if HOLDFLAG negative).

DELTEM$\underline{P}$: Value of CDU angle change, converted to body rates, computed in "RCSATT", scale factor B-1, units revolutions/deci-second. It is computed double precision.

DF$\underline{T}$ (DFT_0, DFT_1, DFT_2): Single precision equivalent times, scale factor B10, units centi-seconds, giving the equivalent single-jet burning time computed for roll, pitch, and yaw axes respectively, for use in the rate filter (in "RCSATT") to update body-angle rates.

DRH$\underline{O}$ ($DRHO_0$, $DRHO_1$, $DRHO_2$): Value of body rate used as input to ADO$\underline{T}$ filter, scale factor B-1, units revolutions/deci-second, computed in "RCSATT" as the difference between measured (from CDU data) and ADO$\underline{T}$ body rates, plus a weighted estimate of previous DRH$\underline{O}$.

EDOT: Value of attitude rate error used in "JLOOP", scale factor B-3, units revolutions/deci-second (set to $ADOT_i$, decremented by $WBODY_i$ if HOLDFLAG $\neq$ 0). Absolute value stored in EDOTVEL.

ERRO$\underline{R}$ ($ERROR_0$, $ERROR_1$, $ERROR_2$): Single precision value of attitude error for automatic maneuvers or attitude hold (computed in "AHFNOROT"), or RHC maneuvers (loaded with MERRO$\underline{R}$ in "T5PHASE2"), used to load AERR at the start of "JLOOP". Scale factor is B-1, units revolutions. Set 0 if free mode inputs are made, and can be displayed on attitude error needles (A$\underline{K}$ = - ERRO$\underline{R}$) if bit 9 (NEEDLFLG) of FLAGWRDO = 0 (by e.g. V61 means).

HOLDFLAG: See Digital Autopilot Interface Routines.

IMODES33: See IMU Computations.

Jd$\underline{M}$ (JdM_0, JdM_1, JdM_2): See Digital Autopilot Interface Routines.

K_{14ms}: Single precision constant, program notation "=14MS", scale factor B10, units centi-seconds. Value is 23 x 2^{-14}, corresponding to about 14.4 milliseconds.

K_{afrg}: Single precision constant, program notation "FOUR", scale factor B-1, units revolutions, giving the "flat region" for "JLOOP" (the offset of the dead-zone line when the rate error changes polarity). Value is 4 x 2^{-14}, corresponding to about 0.044°("encourage limit cycle").

K_{dftm_i}: Set of single precision constants, program notation "DFTMAX", scale factor B10, units centi-seconds. Values are:

i	Value	True Value
-3	-480×2^{-14}	-0.3 seconds
-2	-320×2^{-14}	-0.2 seconds
-1	-160×2^{-14}	-0.1 seconds
0	0	0.0 seconds
1	160×2^{-14}	0.1 seconds
2	320×2^{-14}	0.2 seconds
3	480×2^{-14}	0.3 seconds

K_{gn1_i}: Set of single precision constants, program notation "GAIN1", scale factor B0, used in "RCSATT" to obtain a new value of DRHO (from samples taken 10 times/second). Value of index given by contents of ATTKALMN.

i	Value	True Value	
10	0.9342	0.93420	True Value obtained by converting octal memory information to decimal.
9	0.8151	0.81512	
8	0.6933	0.69330	
7	0.5970	0.59698	
6	0.5223	0.52228	
5	0.4634	0.46338	
4	0.4161	0.41608	
3	0.3774	0.37738	
2	0.3452	0.34521	
1	0.3180	0.31799	
0	0.0640	0.06403	(steady-state gain)
-1	0.2112	0.21118	(high-rate automatic/RHC maneuvers)
-2	0.8400	0.84003	(LM-off translation)
-3	0.2112	0.21118	(LM-on translation)

K_{gn2_i}: Set of single precision constants, program notation "GAIN2", scale factor B0, used in "RCSATT" to obtain a new value of ADOT (from DRHO information acquired 10 times/second). Value of index given by contents of ATTKALMN.

i	Value	True Value	
10	0.8683	0.86829	See above note for K_{gn1_i}
9	0.4817	0.48169	
8	0.2955	0.29547	
7	0.1985	0.19849	
6	0.1422	0.14221	
5	0.1069	0.10687	
4	0.0832	0.08319	
3	0.0666	0.06659	
2	0.0545	0.05450	
1	0.0454	0.04541	
0	0.0016	0.00159	
-1	0.0174	0.01740	
-2	0.3600	0.35999	
-3	0.0174	0.01740	

$K_{m75degs}$: Single precision constant, program notation "-75DEGS", scale factor B-1, units revolutions. Nominal value is -0.41666, corresponding to approximately -75°. Octal value is 62524_8, which corresponds (after a one-bit correction reflecting formation of the absolute value) to - 15254_8, or about -75.015° (ignoring 2's complement).

K_{mantab_i} (i = 0-7): Set of single precision constants, program notation "MANTABLE", scale factor B-9, units revolutions/deci-second. The single precision constants are shifted right (double precision) by 6 places for loading into WBODY and by 8 places for decrementing MERROR in "T5PHASE2". Nominal values are:

i	Value	Corresponds to
0	0.0071111	0.05 degree/sec
1	-0.0071111	-0.05 degree/sec
2	0.028444	0.20 degree/sec
3	-0.028444	-0.20 degree/sec
4	0.071111	0.50 degree/sec
5	-0.071111	-0.50 degree/sec
6	0.284444	2.00 degree/sec
7	-0.284444	-2.00 degree/sec

Nominal value times 3600/512 gives value in degrees/sec.

K_{mintu_i}: Set of single precision constants, program notation "MINTAU", scale factor B10, units centi-seconds. Values for i = 0 and i = 3 are zero; for i = 1, value is 23×2^{-14} (about 14.4 ms); and for i = 2, value is -23×2^{-14} (about -14.4 ms).

K_{mpt1sc}: Single precision constant, program notation "=-.1SEC", scale factor B10, units centi-seconds. Value is -160×2^{-14}, corresponding to -0.1 second.

K_{njt_i}: Set of single precision constants, program notation "NJET", scale factor B0, used to correct jet burn times (which are derived based on one-jet operation) for the number of jets actually to be used, and also to compensate for polarity of TAU (negative for negative jets). Values are:

i	Value
-3	-0.333333
-2	-0.5
-1	-0.999999
0	0.0
1	0.999999
2	0.5
3	0.333333

K_{p24}: Single precision constant, program notation "=.24", scale factor B-2, value 0.24. Used in "REDAP" as an initial condition setting for SLOPE. Value corresponds to $(-1) \times -0.6 \times 0.1 \times 2^2$, where first term is an equation factor, second is slope of attitude rate error limit line vs attitude error (in units of (revolutions/second)/revolution), third converts to units of deci-seconds, and fourth is scale factor.

K_{ptlsc}: Single precision constant, program notation "=+.1SEC", scale factor B10, units centi-seconds. Value is 160 x 2^{-14}, corresponding to 0.1 second.

K_{pytab_i}: Single precision set of constants, program notation "PYTABLE", giving information on pitch and yaw jets selected, together with the number of jets used for the function. See information below.

K_{quadan}: Single precision constant, program notation "QUADANGL", scale factor B-1, units revolutions, "used to account for the roll displacements of the reaction jets with respect to navigation base coordinates." Value is 660 x 2^{-14}, corresponding to about 7.251°.

K_{rtab_i}: Single precision set of constants, program notation "RTABLE", giving information on roll jets selected (for AC or BD quads), together with number of roll jets (with polarity). See information below.

K_{sope2}: Single precision constant, program notation "SLOPE2", scale factor B-2, units (revolutions/deci-second)/revolution. Value is 0.32, corresponding to (-1) x -0.8 x 0.1 x 2^2, where the first term is an equation factor, second is "hysteresis slope" of attitude rate error vs attitude error (in units of (revolutions/second)/revolution), third converts to units of deci-seconds, and fourth is scale factor(helps disturbances).

K_{whdsop}: Single precision constant, program notation "WLH/SLOP", scale factor B-1, units revolutions. Octal value is 00114_8, corresponding to about 0.835°. Nominal value is (0.4 + 0.1)/0.6 = 0.833°: first numerator term is nominal "rate limit" of 0.4°/second, second is nominal half-width of 0.1°/second, and denominator is SLOPE.

K_{wim}: Constant, program notation "WL", scale factor B-3, units revolutions/deci-second. Value is 0.0008888888, corresponding to about 0.4°/second (nominal "rate limit").

K_{wimh}: Constant, program notation "WLH", scale factor B-3, units revolutions/deci-second. Value is 0.0011111111, corresponding to about 0.5°/second ("rate limit" plus half-width of 0.1°/sec).

K_{wimmh}: Constant, program notation "WLMH", scale factor B-3, units revolutions/deci-second. Value is 0.0006666666, corresponding to about 0.3°/second ("rate limit" minus half-width of 0.1°/sec).

$K_{wmhdsop}$: Single precision constant, program notation "WL-H/SLP", scale factor B-1, units revolutions. Octal value is 00055_8, corresponding to about 0.494°. Nominal value is (0.4 - 0.1)/0.6 = 0.500°: cf. K_{whdsop} for meaning of terms.

K_{xn1ndx_i}: Set of single precision constants, program notation "XLN1NDX", scale factor B14, used to provide index information from input data on control configuration. Value is equal to i for i = 0, 1, and 2, and is equal to zero for i = 3.

K_{xtndx_i}: Set of single precision constants, program notation "XLNNDX", scale factor B14, used to provide index information for combining translation and rotation commands. Value is equal to (3 i) for i = 0, 1, and 2, and is equal to zero for i = 3.

K_{yztab_i}: Set of single precision constants, program notation "YZTABLE", giving information permitting the combination of Y or Z translations with roll, as well as number of jets giving a "net roll torque". See information below.

KMJ (KMJ_0, KMJ_1, KMJ_2): See Digital Autopilot Interface Routines.

MERROR: Value of integrated rate error used with RHC ("manual") control, scale factor B-1, units revolutions, computed in double precision (a single precision version is loaded into ERROR in "T5PHASE2" if manual RHC inputs are provided). The cells are initialized to 0 in "REDAP" and are also set 0 in "T5PHASE2" if bits 13-11 of RCSFLAGS are not all 0 (i.e. damping not done); they are updated in "RCSATT" by the measured (from CDU first differences) attitude rate, from which is subtracted desired rate (if damping done) in "T5PHASE2".

NPJETS: Single precision value of number of pitch jets to be fired, scale factor B14. Sign of TAU_1 added in "PITCHTIM".

NRJETS: Single precision value of number of roll jets to be fired (with sign indicating polarity), scale factor B14.

NYJETS: Single precision value of number of yaw jets to be fired, scale factor B14. Sign of TAU_2 added in "YAWTIME".

PINDEX: Single precision index parameter, scale factor B14, indicating the polarity of the pitch command desired: 0 if none, 1 if plus pitch, and 2 if minus pitch.

PMANNDX: See Digital Autopilot Interface Routines.

PWORD1, PWORD2: Single precision cells containing in bits 8-1 the pitch-command bits to be loaded in channel 5 when "T6START" is entered and after the pitch time duration has expired, respectively. The pitch time duration originally is in BLAST1+0.

RACFAIL: See Digital Autopilot Interface Routines.

RATEINDX: See Digital Autopilot Interface Routines.

RBDFAIL: See Digital Autopilot Interface Routines.

RCSFLAGS: See Digital Autopilot Interface Routines.

$\underline{\mathrm{RHO}}$ (RHO_0, RHO_1, RHO_2): Value of previous $\underline{\mathrm{CDU}}$ information (initialized in "REDAP") used to compute input to attitude rate filter in "RCSATT", scale factor B-1, units revolutions in twos complement.

RINDEX: Single precision index parameter, scale factor B14, indicating the polarity of the roll command desired: 0 if none, 1 if plus roll, and 2 if minus roll.

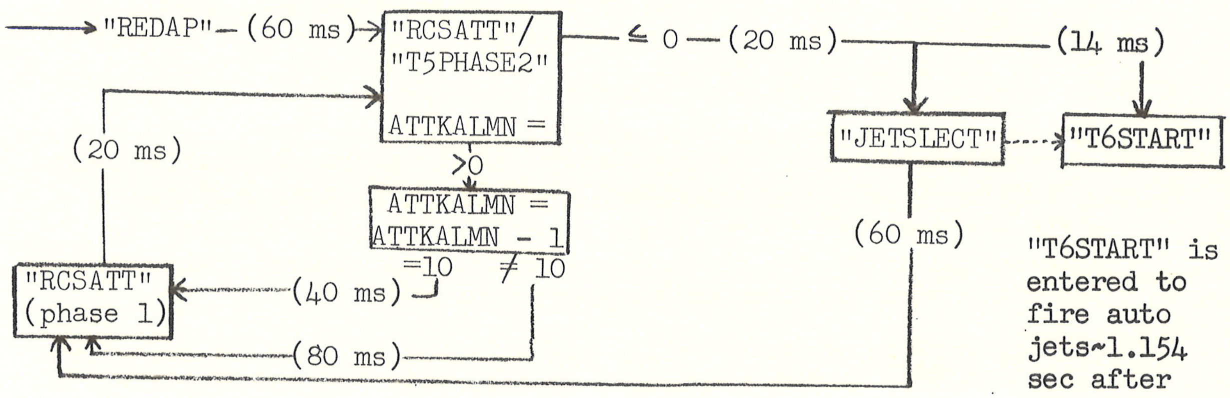
RMANNDX: See Digital Autopilot Interface Routines.

RWORD1, RWORD2: Single precision cells containing in bits 8-1 the roll-command bits (perhaps combined with Y and/or Z axis translation) to be loaded into channel 6 when "T6START" is entered and after the roll time duration has expired, respectively. The roll time duration originally is in BLASTO+0.

SLOPE: Single precision cell, scale factor B-2, units (revolutions/deci-second)/revolution, set to K_{p24} in "REDAP" and not subsequently changed. It is a "variable" (i.e. in erasable memory) because the computer hardware divide order is designed to accept operands only from erasable memory.

T5LOC: See Digital Autopilot Interface Routines.

T5PHASE: Single precision cell, scale factor B14, used to control the program branching that takes place when program interrupt #2 transfers to "RCSATT". A positive non-zero value (of any magnitude) causes the RCS DAP to be initialized; a +0 value causes "T5PHASE2" to be entered; a negative non-zero value (of any magnitude) causes a DAP "restart" (some of the initialization, i.e. HOLDFLAG, is bypassed), as set in "REDORCS"; and a -0 value causes the computations in "RCSATT" to be continued (to perform the rate filtering functions if appropriate). The continuation of "RCSATT" is also known as "phase 1" of the RCS DAP, since the normal sequencing of the RCS DAP is:



"T6START" is entered to fire auto jets~1.154 sec after "REDAP".

T5TIME: Single precision time, scale factor B14, units centi-seconds, used "to compensate for delays" in acting on program interrupt #2, so as to increase the likelihood that the CDU angles for determining attitude rates will be sampled at uniform 0.1-second intervals. It is initialized in "REDAP" to 4, thus causing "RCSATT" (phase 1) to be entered (8 - 4 = 4 cs) after "T5PHASE2" is entered the first time, i.e. 10 cs after $\underline{RHO}$ initialized. The "nominal" contents at other times should be zero (it is reset to TIME5 after "RCSATT" (phase 1) is entered).

T6LOC: See Digital Autopilot Interface Routines.

$\underline{TAU}$ (TAU_0, TAU_1, TAU_2): Single precision values of desired one-jet burn times to accomplish required rotations in roll, pitch, and yaw respectively, scale factor B10, units centi-seconds.

$\underline{THETADX}$ (THETADX, THETADY, THETADZ): Single precision values, scale factor B-1, units revolutions, twos complement, of desired CDU angles in DAP, used to form the attitude errors for "JLOOP". Should not be confused with $\underline{THETAD}$ (components $THETAD_x$ etc), which is a Major Variable.

TIME5, TIME6: See Digital Autopilot Interface Routines.

$\underline{WBODY}$ ($WBODY_0$, $WBODY_1$, $WBODY_2$): Values of desired angular rates, scale factor B-3, units revolutions/deci-second, used to modify the rate error information in "JLOOP" if HOLDFLAG $\neq$ 0. Cells set in "NEWANGL" and "MANUSTOP" (to 0) to desired rates, and in "T5PHASE2" for RHC inputs. Also set in R61. Cells $\underline{OMEGAC}$ for TVC.

XNDX1, XNDX2: Single precision index cells, scale factor B14, set at the start of "JETSLECT" to the complement of bits 8-7 (shifted right 6 places) of channel 31, the X-axis translation hand controller input. XNDX1 is the "AC Quad" translation index (reset 0 if XTRANS = 1, and used in conjunction with "pitch" information) and XNDX2 is the "BD Quad" translation index (reset 0 if XTRANS = -1, and used in conjunction with "yaw" information).

XTRANS: See Digital Autopilot Interface Routines.

YINDEX: Single precision index parameter, scale factor B14, indicating the polarity of the yaw command desired: 0 if none, 1 if plus yaw, and 2 if minus yaw.

YMANNDX: See Digital Autopilot Interface Routines.

YNDX: Single precision index cell, scale factor B14, set at the start of "JETSLECT" to the complement of bits 10-9 (shifted right 8 places) of channel 31, the Y-axis translation hand controller input.

YWORD1, YWORD2: Single precision cells containing in bits 8-1 the yaw-command bits to be loaded in channel 5 when "T6START" is entered and after the yaw time duration has expired, respectively. The yaw time duration originally is in BLAST2+0.

ZNDX: Single precision index cell, scale factor B14, set at the start of "JETSLECT" to the complement of bits 12-11 (shifted right 10 places) of channel 31, the Z-axis translation hand controller input.

Value of Jet Table Data

Information on pitch and yaw jets is stored in K_{pytab_i}, where i is in the range 0 - 14. Bits 12-11 give the number of yaw jets used to perform the rotation, and bits 10-9 give the number of pitch jets used. Bits 8-5 give the yaw output bits (to be loaded in the corresponding bit positions of channel 5), while bits 4-1 give the pitch output bits (to be loaded into the corresponding bits of channel 5).
The value of i depends on the failure information (or, if no failures, the X-translation requested) as well as the rotation desired.

$i = 3a + b$

a = 0 for no X-translation
= 1 for no failures, +X translation
= 2 for no failures, -X translation
= 3 for A (pitch) or B (yaw) failures
= 4 for C (pitch) or D (yaw) failures

b = 0 for no desired rotation
= 1 for + desired rotation
= 2 for - desired rotation

(for PWORD2 and YWORD2, b = 0; a = 0 if any failures)

i	K_{pytab}	# of Yaw	Yaw Jet #	# of Pitch	Pitch Jet #
0	0000_8	0	---	0	---
1	5125_8	2	7,5	2	3,1
2	5252_8	2	6,8	2	2,4
3	0231_8	0	6,5	0	2,1
4	2421_8	1	5	1	1
5	2610_8	1	6	1	2
6	0146_8	0	7,8	0	3,4
7	2504_8	1	7	1	3
8	2442_8	1	8	1	4
9	0000_8	0	---	0	---
10	2421_8	1	5	1	1
11	2442_8	1	8	1	4
12	0000_8	0	---	0	---
13	2504_8	1	7	1	3
14	2610_8	1	6	1	2

Jet #	Bit	Quad	Rotation	Translation
1	1	C	+ Pitch	+X
2	4	A	- Pitch	+X
3	3	A	+ Pitch	-X
4	2	C	- Pitch	-X
5	5	D	+ Yaw	+X
6	8	B	- Yaw	+X
7	7	B	+ Yaw	-X
8	6	D	- Yaw	-X

Information on roll jets (for rotation combined with translation) is stored in K_{rtab_i}, where i is in the range 0 - 14. Bits 14-12 give the number of roll jets for using Quads B/D, while bits 11-9 give the number for using Quads A/C (the number in both cases is given by the value of the bits - 2, thus providing sign information). Bits 8-5 give the Quad A/C output (to be loaded into the corresponding bit positions of channel 6), while bits 4-1 give the Quad B/D output (likewise to be loaded into the corresponding bit positions of channel 6). The value of i is defined as for K_{pytab_i}, except that Y translation is associated with Quad A/C and the Z translation with Quad B/D.

i	K_{rtab}	# of A/C	A/C Jet #	# of B/D	B/D Jet #
0	11000_8	0	---	0	---
1	22125_8	2	15,13	2	11,9
2	00252_8	-2	14,16	-2	10,12
3	11231_8	0	14,13	0	10,9
4	15421_8	1	13	1	9
5	04610_8	-1	14	-1	10
6	11146_8	0	15,16	0	11,12
7	15504_8	1	15	1	11
8	04442_8	-1	16	-1	12
9	11000_8	0	---	0	---
10	15504_8	1	15	1	11
11	04610_8	-1	14	-1	10
12	11000_8	0	---	0	---
13	15421_8	1	13	1	9
14	04442_8	-1	16	-1	12

The information in K_{yztab_i} is in the same format as K_{rtab}, except that i = 3 c + d, where c = 0 for no failures; c = 1 for A or B failures; and c = 2 for C or D failures. The quantity d = 0 for no translation; d = 1 for + translation; and d = 2 for - translation.

i	K_{yztab}	# of A/C	A/C Jet #	# of B/D	B/D Jet #
0	11000_8	0	---	0	---
1	11231_8	0	14,13	0	10,9
2	11146_8	0	15,16	0	11,12
3	11000_8	0	---	0	---
4	04610_8	-1	14	-1	10
5	15504_8	1	15	1	11
6	11000_8	0	---	0	---
7	15421_8	1	13	1	9
8	04442_8	-1	16	-1	12

The "# of A/C" and "# of B/D" information is used to check that the resulting roll torque on the vehicle is not forced to be zero when the desired roll rotation is non-zero. The K_{yztab} information permits the Quad not used for roll rotation to be used to satisfy translation commands.

The following roll-jet assignments are made (channel 6):

Jet #	Bit	Quad	Rotation	Translation
9	1	B	+ Roll	+Z
10	4	D	- Roll	+Z
11	3	D	+ Roll	-Z
12	2	B	- Roll	-Z
13	5	A	+ Roll	+Y
14	8	C	- Roll	+Y
15	7	C	+ Roll	-Y
16	6	A	- Roll	-Y

TVCDAPON Entered about 0.41 seconds after "IGNITION" or if restart (from "CMDSOUT" for TVCPHASE = -1)

TTMP1 = 0

YDELOFF = 0

PDELOFF = 0

DELYBAR = 0

DELPBAR = 0

YERRB = 0

PERRB = 0

TVCDUMMY = 0 (not used)

TEMREG = 0

ROLLWORD = 0

ROLLFIRE = 0

YTMPi = 0 (i = 6 - 1)

PTMPi = 0 (i = 6 - 1)

OMEGAB = 0

$OMEGAC_z = 0$

$OMEGAC_y = 0$

T5LOC = "TVCINIT1"

Set TIME5 to cause program interrupt #2 in 0.01 second

Resume

TVCINIT1 Entered about 0.42 seconds from start of "IGNITION"

Perform "MASSPROP"

j = (bit 14 of DAPDATR1) (1 for LM on, 0 for LM off)

If $j = 1$: (LM on)

$N10_i = C_{hbn10_i}$ $(i = 0 - 14)$

If $j = 0$:

$N10_i = K_{csmn10_i}$ $(i = 0 - 14)$

$KTLXdI = C_{ektlxdi_j}$

Perform "S40.15"

If $j = 1$:

$TS_1 = K_{8csd2}$

If $j = 0$:

$TS_1 = K_{4csd2}$

$T5TVCDT = 16384 - TS_1$

Set bit 15(SWTOVER) of FLAGWRD9 = 0

$KPRIMEDT = 2\ C_{ekprime_j}\ TS_1$ (the 2 is due to scaling)

$REPFRAC = C_{erepfrac_j}$

$CNTR = K_{tcorr_j}$

VCNTR = 19

$PDELOFF_{sp} = PACTOFF$

PCMD = PACTOFF

$DELPBAR_{sp} = PACTOFF$

$YDELOFF_{sp} = YACTOFF$

YCMD = YACTOFF

$DELYBAR_{sp} = YACTOFF$

If bit 13 of DAPDATR1 = 1: (LM off)

If $|K_{ldattlim}\ ERRBTMP{+}i| \geqslant 2^{-14}$: (i = 0, 1: loaded with $-(ERROR_1, ERROR_2)$ in "IGNITION")

$ERRBTMP{+}i = K_{attlim}\ \mathrm{sgn}\ ERRBTMP{+}i$

(If bit 13 of DAPDATR1 = 1):

$PERRB_{sp}$ = ERRBTMP+0

$YERRB_{sp}$ = ERRBTMP+1

Set bit 3 of RCSFLAGS = 1

Perform "NEEDLER"

Proceed to "TVCINIT4"

<u>TVCINIT4</u> Can be entered from "CMDSOUT" if restart

TVCPHASE = +0

OGANOW = CDU_x

Call "TVCEXEC" in 0.51 seconds

T5LOC = "DAPINIT"

TIME5 = T5TVCDT

Resume

<u>DAPINIT</u> Entered about (0.42 + $\frac{1}{2}$ DAP cycle) sec after start of "IGNITION"

TIME5 = T5TVCDT + (T5TVCDT - 16384) (gives interrupt in one DAP cycle)

T5LOC = "PITCHDAP" (entered about (0.42 + 1.5 DAP cycle) sec after start of "IGNITION": for LM-off, 0.48 second)

PCDUYPST = CDU_y

YCDUYPST = PCDUYPST

PCDUZPST = CDU_z (Note that on first pass yaw rates 50% too big)

YCDUZPST = PCDUZPST

Resume

<u>SWICHOVR</u> Entered from "STABLISH" or "CMDSOUT"

Inhibit interrupts (released when return to caller)

PHASETMP = TVCPHASE

TVCPHASE = -2

YTMPi = 0 (i = 6 - 1)

PTMPi = 0 (i = 6 - 1)

Set bit 15(SWTOVER) of FLAGWRD9 = 1

KTLXdI = $C_{ektlxdi_2}$

Perform "S40.15" (starting at second line)

KPRIMEDT = $K_{fkprimdt}$

REPFRAC = $K_{frepfrac}$

PDELOFF = DELPBAR

YDELOFF = DELYBAR

$N10_i = K_{lbn10_i}$ (i = 0 - 14)

TVCPHASE = PHASETMP

Return

<u>TVCEXEC</u> Entered first about 0.93 second after "IGNITION" (see "TVCINIT4")

If bits 15-14 (DAPBIT1, DAPBIT2) of FLAGWRD6 $\neq 10_2$:

TVCEXPHS = 0

End of task

Call "TVCEXEC" in 0.5 seconds

OGAPAST = OGANOW

OGANOW = CDU_x

AK_0 = OGAD - OGANOW (ones complement difference formed)

OGA = - AK_0, rescaled to scale factor B0 revolutions

Call "ROLLDAP" in 0.03 seconds

Perform "NEEDLER"

If VCNTR > 0:

VCNTR = VCNTR - 1, limited $\geq$ +0 (Note lack of restart protection)

Proceed to "1SHOTCHK"

Perform "FIXCW" (VCNTR was +0)

Perform "S40.15"

VCNTR = 19

Proceed to "1SHOTCHK"

<u>1SHOTCHK</u>

If CNTR > 0:

TS_1 = CNTR - 1

TVCEXPHS = 4

Proceed to "CNTRCOPY"

If CNTR < 0: (Tag here "REPCHEK")

If REPFRAC $\leq$ 0: (Set -0 in "S40.8" if low thrust)

TVCEXPHS = 0

End of task

TS_2 = REPFRAC

TVCEXPHS = 2

Proceed to third line of "TEMPSET"

If bit 7(ENGONFLG) of FLAGWRD5 = 0: (CNTR = 0; Tag "1SHOTOK")

TVCEXPHS = 0

End of task

TVCEXPHS = TVCEXPHS + 1 (sets to 1)

Proceed to "TEMPSET"

TEMPSET

$TS_2 = K_{fcorfrac}$

TVCEXPHS = TVCEXPHS + 1 (sets to 2)

If bit 13 of DAPDATR1 = 1: (means LM-off)

$TS_2 = 2\ TS_2$

$TS_1 = -1$

$TS_3 = PDELOFF + TS_2\ (DELPBAR - PDELOFF)_{sp}$

$TS_4 = YDELOFF + TS_2\ (DELYBAR - YDELOFF)_{sp}$

TVCEXPHS = TVCEXPHS + 1 (sets to 3)

PDELOFF = TS_3

PACTOFF = $TS_{3_{sp}}$

YDELOFF = TS_4

YACTOFF = $TS_{4_{sp}}$

TVCEXPHS = TVCEXPHS + 1 (sets to 4)

Proceed to "CNTRCOPY"

CNTRCOPY

CNTR = TS_1

TVCEXPHS = 0

End of task

PITCHDAP Entered initially due to logic in "DAPINIT"; then from "YAWDAP" logic.

T5LOC = "YAWDAP"

TIME5 = T5TVCDT

Set PCDUYPST = CDU_y and TS = PCDUYPST

MCDUYDOT = TS - PCDUYPST (ones complement difference formed)

If $|K_{1drtim}\ \text{MCDUYDOT}| \geq 2^{-14}$:

MCDUYDOT = 0

Set PCDUZPST = CDU_z and TS = PCDUZPST

MCDUZDOT = TS - PCDUZPST (ones complement difference formed)

If $|K_{1drtim}\ \text{MCDUZDOT}| \geq 2^{-14}$:

MCDUZDOT = 0

ERRBTMP = PERRB + $OMEGAC_y$

$OMEGAB_y$ = - MCDUYDOT COSCDUZ COSCDUX - MCDUZDOT SINCDUX

ERRBTMP = ERRBTMP - $OMEGAB_y$

If $|K_{1derrim}\ \text{ERRBTMP}| \geq 2^{-14}$:

$ERRBTMP_{sp}$ = K_{errim} sgn ERRBTMP

TMP1 = PTMP1

TMP3 = PTMP3

TMP5 = PTMP5

Perform "FWDFLTR"

CMDTMP = PDELOFF + CMDTMP

Round CMDTMP to single precision

If $|K_{1dactsat}\ \text{CMDTMP}| \geq 2^{-14}$:

CMDTMP = K_{actsat} sgn CMDTMP

TVCPITCH = TVCPITCH + CMDTMP - PCMD

Set bit 11 (Gate output from TVCPITCH) of channel 14 = 1

TMP2 = PTMP2

TMP4 = PTMP4

TMP6 = PTMP6

Perform "PRECOMP"

DELBRTMP = K_{emat} DELPBAR + $K_{onemeat}$ CMDTMP

Perform "PCOPY"

Resume

PCOPY

TVCPHASE = TVCPHASE + 1 (sets to 1)

PTMPi = TMPi (i = 1 - 6)

PERRB = ERRBTMP

$AK_1 = ERRBTMP_{sp}$

PCMD = CMDTMP

DELPBAR = DELBRTMP

TVCPHASE = TVCPHASE + 1 (sets to 2)

Return

YAWDAP Entered due to "PITCHDAP" logic.

T5LOC = "PITCHDAP"

TIME5 = T5TVCDT

Perform the following for i = Y(or y) and then i = Z (or z):

Set YCDUiPST = CDU_i and TS = YCDUiPST

MCDUiDOT = TS - YCDUiPST (ones complement difference formed)

If $|K_{1drtim}\ MCDUiDOT| \geq 2^{-14}$:

MCDUiDOT = 0

$ERRBTMP = YERRB + OMEGAC_z$

$OMEGAB_z$ = MCDUYDOT COSCDUZ SINCDUX - MCDUZDOT COSCDUX

$ERRBTMP = ERRBTMP - OMEGAB_z$

If $|K_{1derrim}\ ERRBTMP| \geq 2^{-14}$:

$ERRBTMP_{sp} = K_{errim}$ sgn ERRBTMP

TMP1 = YTMP1

TMP3 = YTMP3

TMP5 = YTMP5

Perform "FWDFLTR"

CMDTMP = YDELOFF + CMDTMP

Round CMDTMP to single precision

If $|K_{1dactsat}\ \text{CMDTMP}| \geq 2^{-14}$:

$\text{CMDTMP} = K_{actsat}\ \text{sgn CMDTMP}$

TVCYAW = TVCYAW + CMDTMP - YCMD

Set bit 12(Gate output from TVCYAW) of channel 14 = 1

TMP2 = YTMP2

TMP4 = YTMP4

TMP6 = YTMP6

Perform "PRECOMP"

$\text{DELBRTMP} = K_{emat}\ \text{DELYBAR} + K_{onemeat}\ \text{CMDTMP}$

Perform "YCOPY"

Resume

YCOPY

TVCPHASE = TVCPHASE + 1 (sets to 3)

YTMPi = TMPi (i = 1 - 6)

YERRB = ERRBTMP

$AK_2 = \text{ERRBTMP}_{sp}$

YCMD = CMDTMP

DELYBAR = DELBRTMP

TVCPHASE = 0

Return

FWDFLTR

$\text{DAP1} = \text{N10}_0\ \text{ERRBTMP} + \text{TMP1}$ (N10_0 is "N10")

$\text{DAP2} = \text{N10}_5\ \text{DAP1} + \text{TMP3}$ (N10_5 is "N20")

If bit 14 of DAPDATR1 = 1: (LM attached)

$\text{DAP3} = \text{N10}_{10}\ \text{DAP2} + \text{TMP5}$ (N10_{10} is "N30")

If bit 14 of DAPDATR1 = 0: (LM off)

DAP3 = DAP2 (rescaled to scale factor B-3 revolutions)

CMDTMP = - VARK DAP3

Return

PRECOMP

$TMP1 = N10_1$ ERRBTMP $- N10_3$ DAP1 + TMP2 ($N10_1$ is "N11" and $N10_3$ is "D11")

$TMP2 = N10_2$ ERRBTMP $- N10_4$ DAP1 ($N10_2$ is "N12" and $N10_4$ is "D12")

$TMP3 = N10_6$ DAP1 $- N10_8$ DAP2 + TMP4

$TMP4 = N10_7$ DAP1 $- N10_9$ DAP2

If bit 13 of DAPDATR1 = 0: (LM on)

$TMP5 = N10_{11}$ DAP2 $- N10_{13}$ DAP3 + TMP6

$TMP6 = N10_{12}$ DAP2 $- N10_{14}$ DAP3

Return

REDOTVC

If TVCEXPHS > 0:

Call "EXRSTRT" in 0.09 seconds

If TVCPHASE < 0:

If bit 2 of TVCPHASE = 1: (i.e. -1, not -2)

PCMD = PACTOFF

YCMD = YACTOFF

If TVCPHASE = 1:

Perform "PCOPY" (starting at second line)

If TVCPHASE = 3:

Perform "YCOPY" (starting at second line)

Set bits 11(Disengage Optics DAC) and 8(TVC Enable) of channel 12 = 1

T5LOC = "ENABL2"

Set TIME5 to cause program interrupt #2 in 0.06 seconds

Resume

ENABL2

Set bit 2(Enable Optics CDU Error Counters) of channel 12 = 1

T5LOC = "CMDSOUT"

Set TIME5 to cause program interrupt #2 in 0.02 seconds

Resume

CMDSOUT

TVCPITCH = PCMD - 0 (avoids loading with +0)

TVCYAW = YCMD - 0 (avoids loading with +0)

Set bits 12-11 (Gate outputs from TVCPITCH & TVCYAW) of channel 14 = 1

If TVCPHASE < 0:

If |TVCPHASE| - 1 = 0:

Proceed to "TVCDAPON"

Perform "SWICHOVR" (starting at 4th line)

Proceed to "TVCINIT4"

EXRSTRT

Proceed to appropriate point in "TVCEXEC" routine (just after the point where TVCEXPHS was incremented to its present value)

ROLLDAP Called by "TVCEXEC" every 0.5 second (0.03 second after start of "TVCEXEC")

$OGARATE = K_{rrt}\ (OGANOW - OGAPAST)$ (ones complement difference)

If |ROLLFIRE| = 0:

If |TEMREG| ≠ 0:

TEMREG = 0

End of task

$TS = K_{db} - \frac{1}{2}\ 1dCONACC\ \ OGARATE^2$

SGNRT = 1 sgn OGARATE (0 considered negative)

DELOGA = OGA − TS SGNRT

SGNDOG = 1 sgn DELOGA   (0 considered negative)

If SGNRT sgn SGNDOG $\geqslant$ 0:

  DELOGA = DELOGA + K_{db} sgn SGNDOG

  DELOGART = − K_{mcrat} sgn SGNDOG

  Proceed to "ONROLL"

If OGARATE sgn SGNDOG + K_{maxim} $\leqslant$ 0:

  RATDES = − K_{maxim} sgn SGNDOG

  Proceed to "ROLLSET"

TS = (K_{msope} OGA + OGARATE) sgn SGNDOG − $K_{intercp}$

If TS $\leqslant$ 0, proceed to "NOROLL"

If OGARATE sgn SGNDOG + K_{minim} $\leqslant$ 0, proceed to "NOROLL"

If OGA sgn SGNDOG − K_{db} $\leqslant$ 0, proceed to "NOROLL"

DELOGA = OGA

DELOGART = OGARATE

Proceed to "ONROLL"

<u>ONROLL</u>

TS_1 = K_{msope} ldCONACC DELOGART − 1 sgn SGNDOG   (scaled B4)

TS_2 = K_{msope} ldCONACC DELOGART^2 + K_{msope} (DELOGA sgn SGNDOG − K_{db})

  + K_{mcrat} (scaled B0)

If $|TS_2| - 2^{-4}|TS_1| < 0$:   (implemented by division of TS_2 shifted right 14 places by TS_1 and check of A: the 2^{-4} comes about from scale factors)

  RATDES = TS_2 / TS_1

If $|TS_2| - 2^{-4}|TS_1| \geqslant 0$:

  RATDES = +MAX sgn (TS_2 / TS_1)   (magnitude $\cong 22\frac{1}{2}^{\circ}$/sec)

Proceed to "ROLLSET"

ROLLSET

TEMREG = RATDES (same cell)

TS = RATDES - OGARATE

If $|TS| \geq 2^{-4}$ rev/sec:

$TS = (2^{-4} - 2^{-18})$ sgn TS rev/sec

$TS = \frac{1}{2} K_{t6sc}$ 1dCONACC TS, modulo 2^{11} centi-seconds

TEMREG = 2 TS

If $|TEMREG| \geq 2^{10}$ centi-seconds:

$TEMREG = (2^{10} - 2^{-4})$ sgn TEMREG centi-seconds

If TEMREG = 0:

Proceed to "NOROLL"

TS_1 = TEMREG ROLLFIRE

If $TS_1 < 0$, proceed to "NOROLL"

If $|TS_1| = 0$:

If $|TEMREG| - K_{tminfr} \leq 0$:

Proceed to "NOROLL"

If $TS_1 > 0$:

SGNDOG = 0 (used as flag in "JETROLL")

If $|K_{1dtmxfir}\ TEMREG| \geq 2^{-14}$:

$TEMREG = K_{tmxfir}$ sgn TEMREG

ROLLFIRE = +MAX sgn TEMREG

TEMREG = |TEMREG|

If SGNDOG $\neq$ 0, proceed to "JETROLL"

If (TIME6 - TEMREG) $\leq$ 0, End of task (don't extend burn duration)

Proceed to "JETROLL"

NOROLL

ROLLFIRE = -0

TEMREG = -0

Proceed to "JETROLL"

JETROLL

T6LOC = "NOROLL1"

TIME6 = TEMREG

If SGNDOG = 0: (set in "ROLLSET" to continue present jets)

End of task

If ROLLFIRE = 0:

Set bit 15(TIME6 Count Enable) of channel 13 = 1

End of task

If ROLLFIRE > 0:

If bit 1 of ROLLWORD = 1:

Set bit 1 of ROLLWORD = 0

Channel 6 = K_{pro2}

Set bit 15(TIME6 Count Enable) of channel 13 = 1

End of task

Set bit 1 of ROLLWORD = 1

Channel 6 = K_{pro1}

Set bit 15(TIME6 Count Enable) of channel 13 = 1

End of task

If bit 2 of ROLLWORD = 1: (ROLLFIRE < 0)

Set bit 2 of ROLLWORD = 0

Channel 6 = K_{mro2}

Set bit 15(TIME6 Count Enable) of channel 13 = 1

End of task

Set bit 2 of ROLLWORD = 1

Channel 6 = K_{mrol}

Set bit 15(TIME6 Count Enable) of channel 13 = 1

End of task

<u>NOROLL1</u>

ROLLFIRE = +0

Channel 6 = 0

Resume

Quantities in Computations

See also list of major variables and list of routines

1dCONACC: See Digital Autopilot Interface Routines.

AK_0, AK_1, AK_2: See Digital Autopilot Interface Routines.

$C_{ekprime_0}$: Single precision erasable memory constant, program notation "EKPRIME", scale factor B-5, units (revolutions/centi-second)/radian, giving the "steer law gain" for LM-off (used in "TVCINIT1" to initialize KPRIMEDT for "S40.8"). The maximum value that can be stored in the cell is $(100 \times 2\pi \times 2^{-5}) = (100\pi/16)$ radians/sec per radian, where "maximum value" actually is one least increment more than 37777_8.

$C_{ekprime_1}$: Single precision erasable memory constant, program notation "EKPRIME +1", scale factor B-5, units (revolutions/centi-second)/radian, giving the "steer law gain" set in "TVCINIT1" for LM-on (see $C_{ekprime_0}$ and $K_{fkprimdt}$).

$C_{ektlxdi_0}$: Single precision erasable memory constant, program notation "EKTLX/I", used in "TVCINIT1" to initialize KTLXdI for "S40.15", for the LM-off configuration. Scale factor is B4, units ASCREV/sec^2, where one ASCREV = 1.07975111 revolutions (see K_{actsat}). To convert to memory information, value in units of $(1/sec^2)$ should be divided by (1.07975111×2^4), giving a fraction which in turn is placed in the cell. Alternatively, cell could be considered scaled B18 in units of CDU actuator pulses per revolution times (newton-meters/kg-$meters^2$), where second term compensates for the multiplication by IAVG/TLX (and reduces to $1/sec^2$, of course). There are 2^{14} actuator pulses in 1.07975111 revolutions.

$C_{ektlxdi_1}$: Single precision erasable memory constant, program notation "EKTLX/I +1", used in "TVCINIT1" to initialize KTLXdI for "S40.15" for LM on (see also $C_{ektlxdi_2}$). Scale factor is B2, units ASCREV/sec^2 (see $C_{ektlxdi_0}$).

$C_{ektlxdi_2}$: Single precision erasable memory constant, program notation "EKTLX/I +2", used in "SWICHOVR" to initialize KTLXdI for "S40.15" for "low bandwidth" LM on. Scale factor is B2, units ASCREV/sec^2 (see $C_{ektlxdi_0}$).

$C_{erepfrac_0}$: Single precision erasable memory constant, program notation "EREPFRAC", scale factor B2, used as LM-off initialization for REPFRAC in "TVCINIT1" and "S40.8" (note that TS_2, i.e. REPFRAC, is doubled in "TEMPSET" if LM-off, hence effective "system" scale factor is B3).

$C_{erepfrac_1}$: Single precision erasable memory constant, program notation "EREPFRAC +1", scale factor B2, used as LM-on initialization for REPFRAC in "TVCINIT1" and "S40.8". See also $K_{frepfrac}$.

C_{hbn10_i} (i = 0 - 14): Set of single precision erasable memory constants used in "TVCINIT1" to initialize $N10_i$ for LM-on (if "SWICHOVR" entered, $N10_i$ loaded with K_{1bn10_i} instead).

i	Notation	Scale Factor
0	"HBN10"	B0
1	"HBN11/2"	B1
2	"HBN12"	B0
3	"HBD11/2"	B1
4	"HBD12"	B0
5	"HBN20"	B0
6	"HBN21/2"	B1
7	"HBN22"	B0
8	"HBD21/2"	B1
9	"HBD22"	B0
10	"HBN30"	B0
11	"HBN31/2"	B1
12	"HBN32"	B0
13	"HBD31/2"	B1
14	"HBD32"	B0

CMDTMP: Value of required output information computed initially in "FWDFLTR" double precision, and subsequently rounded to single precision in "PITCHDAP" or "YAWDAP", scale factor B0, units ASCREV (or B14 in units of actuator pulses): VARK converts from units of revolutions in "FWDFLTR".

CNTR: Single precision counter, scale factor B14, units counts (of ½ second each), preset in "TVCINIT1" and decremented in "1SHOTCHK". When reduced to 0, the next entrance to "TVCEXEC" causes a "one-shot" correction to PACTOFF and YACTOFF (as well as PDELOFF and YDELOFF), using $K_{fcorfrac}$. CNTR is set to +MAX in "S40.81"when the cutoff decision is made, to inhibit further trim updates after that time (a value of -1 is set in "TEMPSET"/"CNTRCOPY" to signify that the one-shot correction has been made). Since K_{tcorr_1} (initial condition for LM-on of CNTR) is zero, no decrementing of CNTR is required for that configuration.

COSCDUX, COSCDUZ: See Coordinate Transformations. Updated once a second while "CLOCKJOB" is entered, which includes period of time when TVC DAP is on.

DAP1, DAP2, DAP3: Filter quantities computed in "FWDFLTR", scale factor B-1, units revolutions. If LM off, then DAP3 is set equal to DAP2 (otherwise, it is derived), scaled B-3.

DAPDATR1: See Digital Autopilot Interface Routines.

DELBRTMP: Double precision erasable memory cell used to contained the updated value for DELPBAR or DELYBAR (for restart protection purposes), scale factor B14, units CDU actuator output pulses (or scale factor B0 in ASCREV units).

DELOGA: Single precision value of roll attitude error used in "ONROLL", scale factor B0, units revolutions (computed initially in "ROLLDAP" as "distance from switch parabola").

DELOGART: Single precision value of roll attitude rate error used in "ONROLL", scale factor B-4, units revolutions/second.

DELPBAR: Double precision value of estimated "c.g. offset relative to electrical null" in pitch, scale factor B14 in units of CDU actuator pulses (or scale factor B0 in ASCREV units). Initialized to PACTOFF in "TVCINIT1" and updated in "PCOPY" with DELBRTMP (for restart protection purposes). After CNTR reduced to zero, used in "TEMPSET" to updated PDELOFF and PACTOFF (using $K_{fcorfrac}$ and then REPFRAC for weighting). Used in "SPSOFF" to load PACTOFF if suitable initialization completed (i.e. "TVCINIT1" completed, initializing DELPBAR), and no thrust failure.

DELYBAR: Double precision value of estimated "c.g. offset relative to electrical null" in yaw, scale factor B14, units CDU actuator output pulses (or scale factor B0 in ASCREV units). See DELPBAR (substitute yaw axis for pitch axis).

ERRBTMP: Value of pitch or yaw error in body coordinates, scale factor B-1, units revolutions, limited to K_{errim}. Could also be considered to be the "integral of body-axis rate error". Value is displayed on FDAI error needles (due to loading of AK_1 or AK_2 by appropriate copy routine). Loaded in "IGNITION" with RCS DAP pitch and yaw negative (single precision) error information for use in "TVCINIT1", after limiting, to initialize PERRB and YERRB for LM-off.

$K_{1dactsat}$: Single precision constant, program notation "1/ACTSAT", scale factor B0, octal value 00101_8, corresponding to 0.003967285. Effect in equations such as to cause an answer of one least increment (2^{-14} in programmed equations) for an argument of 253 least increments or more (see K_{actsat}).

$K_{1dattlim}$: Single precision constant, program notation "1/ATTLIM", scale factor B0, octal value 00170_8. Effect in equations such as to cause an answer of one least increment (2^{-14} in programmed equations) for an argument of 137 least increments (at B-1 rev., about 1.505^o): see K_{attlim}.

$K_{1derrim}$: Single precision constant, program notation "1/ERRLIM", scale factor B0, octal value 00004_8. Effect in equations such as to cause an answer of one least increment (2^{-14} in programmed equations) for an argument of 2^{-2} or more (see K_{errim}): the 2^{-2} would correspond to 2^{-3} revolutions.

K_{1drtim}: Single precision constant, program notation "1/RTLIM", scale factor B0, octal value 00115_8. Effect in equations such as to cause an answer of one least increment (2^{-14} in programmed equations) for an argument of 213 least increments (at B-1 rev., about 2.34°).

$K_{1dtmxfir}$: Single precision constant, program notation "1/TMXFIR", scale factor B0, octal value 00004_8. Effect in equations such as to cause an answer of one least increment (2^{-14} in programmed equations) for an argument of 2^{-2} or more (see K_{tmxfir}): the 2^{-2} would correspond to 2.56 seconds.

K_{4csd2}: Single precision constant, program notation "BIT2", scale factor B14, units centi-seconds, giving the required time interval between DAP computations for different axes for LM-off. Value is 2×2^{-14}, corresponding to 2 centi-seconds: this gives a time interval of 4 centi-seconds from one pitch axis solution to the next.

K_{8csd2}: Single precision constant, program notation "BIT3", scale factor B14, units centi-seconds, giving the required time interval between DAP computations for different axes for LM-on. Value is 4×2^{-14}, corresponding to 4 centi-seconds: this gives a time interval of 8 centi-seconds from one pitch axis solution to the next.

K_{actsat}: Single precision constant, program notation "ACTSAT", scale factor B14, units CDU actuator output pulses (one pulse is 85.41 seconds). Value is 253×2^{-14}, corresponding to about $6.0024°$. Note that 2^{14} CDU actuator output pulses correspond to about 1.07975111 revolutions; there are about 42.14963 pulses/degree.

K_{attlim}: Single precision constant, program notation "ATTLIM", scale factor B-1, units revolutions. Value is 136 least increments (see $K_{1dattlim}$), or octal 00210_8, corresponding to about 1.494°.

K_{csmn10_i} (i = 0 - 14): Set of single precision constants used in "TVCINIT1" to initialize $N10_i$ for LM-off, program notation (i = 0) "CSMN10". Only values for i = 0 - 9 are significant (since $N10_{10}$ to $N10_{14}$ are employed only for LM-on); values for i = 10 - 14 are the same cells as K_{1bn10_0} to K_{1bn10_4}.

i	Nominal Value	Scale Factor	Nominal True Value		Stored
0	0.99999	B0	0.99999	(N10)	0.99994
1	-0.2549	B1	-0.5098	(N11)	-0.50977
2	0.0588	B0	0.0588	(N12)	0.05878
3	-0.7620	B1	-1.524	(D11)	-1.52405
4	0.7450	B0	0.7450	(D12)	0.74500
5	0.99999	B0	0.99999	(N20)	0.99994
6	-0.4852	B1	-0.9704	(N21)	-0.97046
7	0.0	B0	0.0	(N22)	0.0
8	-0.2692	B1	-0.5384	(D21)	-0.53845
9	0.0	B0	0.0	(D22)	0.0

The "nominal value" is the scaled decimal input to the assembly program; the "stored" value was obtained by converting the octal memory information to decimal and applying the proper scale factor.

K_{db}: Single precision constant, program notation "DB", scale factor B0, units revolutions, giving roll deadband for TVC DAP. Octal value is 00344_8, corresponding to about 5.0098^o.

K_{emat}: Single precision constant, program notation "E(-AT)", scale factor B0, value 37535_8, corresponding to $(1 - K_{onemeat})$.

K_{errim}: Single precision constant, program notation "ERRLIM", scale factor B-1, units revolutions. Value is 10000_8, corresponding to 2^{-3} revolutions or 45^o.

$K_{fcorfrac}$: Single precision constant, program notation "FCORFRAC", scale factor B2. Used in "TEMPSET" as the LM-on c.g. one-shot correction gain (see CNTR above). A value of 2 $K_{fcorfrac}$ is used for LM-off (hence scale factor could be considered B3 for the LM-off application). Value is octal 10000_8, corresponding to 1.0×2^{-2}, where first term is equation value and second is scale factor.

$K_{fkprimdt}$: Single precision constant, program notation "FKPRIMDT", scale factor B-4, units (revolutions/DAP cycle) per radian. Used in "SWICHOVR" to load KPRIMEDT. Value is 0.0102, corresponding approximately to $0.05 \times 0.08 \times (1/2\pi) \times 2^4$, where first term is value of gain in (radians/sec)/radian, second is DAP cycle rate in seconds, third is conversion to revolutions, and 4th is scale factor: see KPRIMEDT.

$K_{frepfrac}$: Single precision constant, program notation "FREPFRAC", scale factor B2, used in "SWICHOVR" and "S40.8" to load REPFRAC for "low-bandwidth" mode of TVC DAP. Nominal value is 0.0375×2^{-2}, where first term is equation value and second is scale factor. The actual stored value corresponds to about 0.0376.

$K_{intercp}$: Single precision constant, program notation "INTERCEP", scale factor B-3, units revolutions. Octal value is 00510_8, corresponding to about $0.2002 \times 45 = 0.9009^o$.

K_{lbn10_i} (i = 0 - 14): Set of single precision constants used in "SWICHOVR" to initialize $N10_i$ for "low bandwidth" LM-on, program notation (i = 0) "LBN10".

i	Nominal Value	Scale Factor	Nominal True Value		Stored
0	0.99999	B0	0.99999	(N10)	0.99994
1	-0.3285	B1	-0.6570	(N11)	-0.65698
2	-0.3301	B0	-0.3301	(N12)	-0.33008
3	-0.9101	B1	-1.8202	(D11)	-1.82019
4	0.8460	B0	0.8460	(D12)	0.84601

i	Nominal Value	Scale Factor	Nominal True Value		Stored
5	0.03125	B0	0.03125	(N20)	0.03125
6	0.0	B1	0.0	(N21)	0.0
7	0.0	B0	0.0	(N22)	0.0
8	-0.9101	B1	-1.8202	(D21)	-1.82019
9	0.8460	B0	0.8460	(D22)	0.84601
10	0.5000	B0	0.5000	(N30)	0.50000
11	-0.47115	B1	-0.9423	(N31)	-0.94226
12	0.4749	B0	0.4749	(N32)	0.47491
13	-0.9558	B1	-1.9116	(D31)	-1.91162
14	0.9372	B0	0.9372	(D32)	0.93719

See note with K_{csmn10_i}.

K_{maxim}: Single precision constant, program notation "MAXLIM", scale factor B-4, units revolutions/second. Value is 0.01388889 x 2^4, corresponding to about 5 degrees/second.

K_{mcrat}: Single precision constant, program notation "LMCRATE", scale factor B-4, units revolutions/second (gives limit cycle roll rate for TVC roll DAP). Value is 00111_8, corresponding to about 0.10025 degree/second.

K_{minim}: Single precision constant, program notation "MINLIM", scale factor B-4, units revolutions/second. Value is 0.00277778 x 2^4, corresponding to about 1 degree/second.

K_{mro1}: Single precision constant, program notation "-ROLL1", octal value 00012_8, corresponding to channel 6 roll jets #10 and #12, giving a SM negative roll using quad B/D.

K_{mro2}: Single precision constant, program notation "-ROLL2", octal value 00240_8, corresponding to channel 6 roll jets #14 and #16, giving a SM negative roll using quad A/C.

K_{msope}: Single precision constant, program notation "-SLOPE", scale factor B0, units (revolutions/second)/revolution. Value is 0.2. Notation arises from fact that "true" slope is negative (an increasing attitude error results in a decreasing attitude rate).

$K_{onemeat}$: Single precision constant, program notation "1-E(-AT)", scale factor B0, octal value 00243_8. Value corresponds to 163/16384, or approximately 0.01, equivalent to the first term in the series expansion of $(1 - e^{-at})$ for at = 0.01.

K_{pro1}: Single precision constant, program notation "+ROLL1", octal value 00005_8, corresponding to channel 6 roll jets #11 and #9, giving a + Roll (for SM). Uses Quad B/D.

K_{pro2}: Single precision constant, program notation "+ROLL2", octal value 00120_8, corresponding to channel 6 roll jets #15 and #13, giving a + Roll (for SM). Uses Quad A/C.

K_{rrt}: Single precision constant, program notation "BIT5", scale factor B11, value 2^{-10}, corresponding to 2 x 2^{-11}, where first term is conversion of rate to units of revolutions/second, and second is scale factor.

K_{t6sc}: Single precision constant, program notation "T6SCALE", scale factor B7, value 31000_8. Value corresponds to 100 x 2^{-7}, where first term converts from seconds to centi-seconds and second is scale factor.

K_{tcorr_0}: Single precision constant, program notation "TCORR", scale factor B14, units counts (used as a preset for CNTR in "TVCINIT1" for LM-off). Value is 5 x 2^{-14}, meaning that the "single-shot" correction will take place the 6th time that "TVCEXEC" is entered, or about 3.43 seconds after "IGNITION" is done.

K_{tcorr_1}: Single precision constant, program notation "TCORR +1", scale factor B14, units counts (used as a preset for CNTR in "TVCINIT1" for LM-on). Value is 0, meaning that the "single-shot" correction will take place the first time that "TVCEXEC" is entered, or about 0.93 seconds after "IGNITION" is done.

K_{tminfr}: Single precision constant, program notation "TMINFIRE", scale factor B10, units centi-seconds. Value is 00030_8, corresponding to 1.5 centi-seconds (i.e. 15 milliseconds).

K_{tmxfir}: Single precision constant, program notation "TMAXFIRE", scale factor B10, units centi-seconds. Value is 250 x 2^{-10}, corresponding to 250 centi-seconds (i.e. 2.5 seconds). See $K_{ldtmxfir}$.

KPRIMEDT: Quantity used in "S40.8" to convert guidance attitude error information into OMEGAC. Scale factor is B-4, units (revolutions/ DAP cycle) per radian. Loaded in "TVCINIT1" and "SWICHOVR". Least significant half expected to be 0 (from "TVCINIT1" loading method).

KTLXdI: Single precision value of appropriate $C_{ektlxdi}$ cell, loaded in "TVCINIT1" or "SWICHOVR" and used in "S40.15". Scale factor is B4 (LM off) or B2 (LM on), program notation "KTLX/I".

MCDUYDOT, MCDUZDOT: Complement of change in CDU_y and CDU_z respectively since previous sample, scale factor B-1, units revolutions/DAP cycle (in general). Computed in "PITCHDAP" and "YAWDAP" based on separate CDU angle samples, and used to compute appropriate component of OMEGA$\underline{B}$. If the magnitude exceeds about 2.33^o, it is set 0 (see K_{ldrtim}). The DAP cycle is 40 ms for LM-off and 80 ms for LM-on. Due to initialization in "DAPINIT", the rate derived the first pass through "YAWDAP" will be 50% too big.

$N10_i$ (i = 0 - 14): Set of filter constants loaded in "TVCINIT1" with K_{csm10_i} (LM-off) or C_{hbn10_i} (LM-on), or in "SWICHOVR" with K_{lbn10_i}. For LM-off, parameters for i = 10 - 14 are not used. Scale factors for i = 1, 3; 6, 8; and 11, 13 are B1, with remainder of constants (all single precision) having scale factor B0.

OGA: Single precision value of roll error (OGANOW - OGAD), scale factor B0, units revolutions, computed in "TVCEXEC". Program notation where computed is "OGAERR".

OGAD: Single precision value of desired outer gimbal angle (i.e. CDU_x), scale factor B-1, units revolutions, set to CDU_x in "IGNITION" (when SPS engine is turned on). Used to cause the roll TVC DAP to maintain "roll attitude hold".

OGANOW: Single precision value of CDU_x, scale factor B-1, units revolutions, sampled in "TVCINIT4" and "TVCEXEC".

OGAPAST: Previous value of OGANOW, scale factor B-1, units revolutions, loaded with previous value of OGANOW in "TVCEXEC".

OGARATE: Single precision value of rate of change of OGANOW computed at the start of "ROLLDAP", scale factor B-4, units revolutions/second. Will be about 2% too big the evaluation after "TVCINIT4" is performed (since delay between successive samples then is 0.51 seconds, not 0.50 seconds).

OMEGA$\underline{B}$: Value of body rate determined from first difference of CDU_y and CDU_z information, scale factor B-1, units revolutions/DAP cycle (computed double precision). The y component is loaded in "PITCHDAP" and the z component in "YAWDAP", program notations "OMEGAYB" and "OMEGAZB" respectively. The "x" component is used for the storage of OGARATE (single precision), and is assigned that tag. The cells used for OMEGA$\underline{B}$ are used in the RCS DAP for ADO$\underline{T}$ (a common cell assignment is made for telemetry considerations). Note that "x" component is B-4 rev/second, not B-1 rev/DAP cycle, and single precision.

OMEGA$\underline{C}$: Value of guidance steering output computed in "S40.8", scale factor B-1, units revolutions/DAP cycle (see KPRIMEDT). The x component is not used; the y component is used in "PITCHDAP"; and the z component is used in "YAWDAP".

PACTOFF: Single precision value of pitch c.g. offset, scale factor B14, units CDU actuator pulses (one pulse is 85.41 seconds). Could also be considered scaled B0 in units of ASCREV (see $C_{ektlxdi_0}$). Can be loaded by R1 of N48 (e.g. in R03). It is measured "relative to electrical null" specified by K_{pbias} (see Steering Computations). Quantity also called "total pitch trim angle." It is used in "TVCINIT1" to initialize filter quantities in TVC DAP, and is updated in "TEMPSET" after the one-shot has taken place (CNTR reduced to 0). Can also be updated in "SPSOFF" (see DELPBAR).

PCDUYPST, PCDUZPST: Previous values of CDU_y and CDU_z for use in deriving MCDUYDOT and MCDUZDOT for pitch DAP, single precision with scale factor B-1, units revolutions, in twos complement. Separate cells are used for yaw DAP (see YCDUYPST). PCDUiPST cells are initialized in "DAPINIT", with "PITCHDAP" entered at a time corresponding to one complete DAP cycle (pitch and yaw) later, giving proper first-difference computation.

PCMD: Single precision value of previous pitch output command information, scale factor B14, units CDU actuator pulses (or B0 in units of ASCREV). It is intended to maintain an image of the information in the optics error counter driven from TVCPITCH output. It is set to PACTOFF in "TVCINIT1" ("S40.6" left engine bell at that position), and updated in "PCOPY" with CMDTMP.

PDELOFF: Value of pitch c.g. offset, scale factor B14, units CDU actuator pulses (or B0 in units of ASCREV). The quantity is a double precision version of PACTOFF (updated in "TEMPSET" and initialized to PACTOFF in "TVCINIT1") used to minimize the effect of computer word length.

PERRB: Value of pitch error in body coordinates, loaded in "PCOPY" with ERRBTMP (which in turn is sum of previous PERRB and ($OMEGAC_y$ - $OMEGAB_y$)), scale factor B-1, units revolutions. Could also be considered to be the "integral of body-axis pitch-rate error", since the $OMEGAi_y$ terms are in units of revolutions/DAP cycle.

PHASETMP: Single precision cell used in "SWICHOVR" to retain the value of TVCPHASE until the computation is complete, to permit setting of TVCPHASE = -2 as an indication to restart logic ("REDOTVC" and "CMDSOUT") that "SWICHOVR" was being performed.

PTMPi (i = 1 - 6): Values of pitch filtered information, initialized to 0 in "TVCDAPON" and "SWICHOVR", and loaded with TMPi in "PCOPY". Scale factors are all B-1, units revolutions.

RATDES: Single precision value of desired roll rate used in "ROLLSET", scale factor B-4, units revolutions/second. Program uses TEMREG cell.

RCSFLAGS: See Digital Autopilot Interface Routines.

REPFRAC: Single precision value of "repetitive c.g. correction fraction", scale factor B2, used in "1SHOTCHK" to control update of the c.g. tracker (if REPFRAC 0 or negative, no update is done) after CNTR becomes 0. It is initialized in "TVCINIT1" to appropriate $C_{erepfrac}$ and (for "long" burns) is loaded in "S40.8" with the same information. If "SWICHOVR" entered, loaded with $K_{frepfrac}$. In "S40.8", is set to -0 if thrust below minimum value, to inhibit updates.

ROLLFIRE: Single precision cell, scale factor B0, set to +MAX for positive roll torque and -MAX for negative roll torque in roll TVC DAP. It is initialized to 0 in "TVCDAPON" and when roll jets are turned off in "NOROLL" or "NOROLL1", and used as a control cell to indicate that roll jet firings have been specified (used at start of "ROLLDAP", with TEMREG, to force at least $\frac{1}{2}$ second of no firing between successive firings).

ROLLWORD: Single precision cell initialized to 0 in "TVCDAPON" and used to permit alternation of the pairs of jets (i.e. quads) used for torque in the TVC roll DAP. Bit 1 is used for positive roll commands and bit 2 for negative roll commands: quad B/D is used if the bit is now 0 and quad A/C if bit is now 1. The bit used for selection is complemented after being used.

SGNDOG: Cell giving sign information for DELOGA (implemented in program by loading a "clear add" order for positive and a "clear subtract" order for negative), program notation "I". It is set 0 in "ROLLSET" if the polarity of the present jet firing is the same as that of desired firing during this evaluation, in order to flag the fact that present jets should be continued (subsequent evaluations can reduce the required roll firing time, but not increase it, due to logic at end of "ROLLSET").

SGNRT: Single precision cell, scale factor B14, giving information on the polarity of OGARATE.

SINCDUX: See Coordinate Transformations (and COSCDUX).

T5LOC: See Digital Autopilot Interface Routines.

T5TVCDT: Single precision value of required setting for TIME5, scale factor B14, units centi-seconds. TIME5 is incremented by hardware means once each centi-second, and when it reaches 16384 (2^{14}), program interrupt #2 is generated. Consequently, to cause program interrupt #2 in 2 centi-seconds from "now", TIME5 is set to 16384 - 2 = 16382; in 4 centi-seconds requires 16380; etc.

T6LOC: See Digital Autopilot Interface Routines.

TEMREG: Single precision cell used for several functions within TVC roll DAP, but left at end with value of time loaded into TIME6, scale factor B10, units centi-seconds. It is preset to 0 in "TVCDAPON", and used with ROLLFIRE at the start of "ROLLDAP" to force at least $\frac{1}{2}$ second of no firing between successive firings.

TIME5: See Digital Autopilot Interface Routines.

TIME6: See Digital Autopilot Interface Routines.

TMPi (i = 1 - 6): Set of cells used to retain PTMPi or YTMPi information, scale factor B-1, units revolutions, in order to achieve restart protection and allow use of the common subroutines "FWDFLTR" and "PRECOMP". All cells double precision.

TTMP1: Double precision cell used for intermediate storage of quantities in TVC DAP (not shown in programmed equations), set 0 in "TVCDAPON" (no functional purpose served by the zeroing).

TVCDUMMY: Single precision cell set 0 in "TVCDAPON", but not otherwise referenced by TVC DAP (cell formerly assigned to a stroking-test variable, and employed for erasable memory layout considerations).

TVCEXPHS: Single precision cell, scale factor B14, used to control the restart logic associated with the TVC DAP for the waitlist task started every half second at "TVCEXEC". It is initialized to 0 in "IGNITION" and at the end of each performance of the "TVCEXEC" task. If it is non-zero when "REDOTVC" is entered, it is used as an indexing parameter to select the appropriate address within the "TVCEXEC" package in "EXRSTRT". For clarity, these addresses are not shown in the programmed equations, but correspond to the line following the one which incremented TVCEXPHS to its present value.

TVCPHASE: Single precision cell, scale factor B14, used to control the restart logic associated with the TVC DAP for the initialization, switchover to low bandwidth (LM-on), pitch copy cycle, and yaw copy cycle (values -1, -2, +1, and +3 respectively). Set to -1 in "IGNITION" and to 0 at the start of "TVCINIT4", to control branching in "REDOTVC". If "SWICHOVR" entered, set to -2, after the present value saved in PHASETMP. The appropriate "PCOPY"/"YCOPY" routine sets to control the pitch/yaw copy cycle restart logic.

TVCPITCH, TVCYAW: See Digital Autopilot Interface Routines.

VARK: Single precision variable gain for TVC pitch and yaw channels, scale factor B4 (LM off) or B2 (LM on), units ASCREV/revolution (or B18/B16 in units of CDU actuator pulses/revolution). It is computed in "S40.15" and used in "FWDFLTR".

VCNTR: Single precision cell, scale factor B14, used for control of the "TVCEXEC" computations. It is normally employed to cause an update of DAP parameters (due to mass change computed every 2 seconds in "S40.8") every 20 entrances to "TVCEXEC" (i.e. every 10 seconds). Since the modifications to the counter are not restart protected, however, a restart in a small interval could cause another modification to the counter (modifying the period, although this should not be a problem). The cell is preset to 19 in "TVCINIT1" and in "TVCEXEC" after updating parameters (when cell reaches +0).

YACTOFF: Single precision value of yaw c.g. offset, scale factor B14, units CDU actuator pulses (or scale B0 in ASCREV). It is measured "relative to electrical null" specified by K_{ybias} (see Steering Computations). See PACTOFF above (updated at analogous times to those for PACTOFF, but of course with yaw parameters).

YCDUYPST, YCDUZPST: Previous values of CDU_y and CDU_z for use in deriving MCDUYDOT and MCDUZDOT for yaw DAP, single precision with scale factor B-1, units revolutions, in twos complement. Separate cells are used for pitch DAP (see PCDUYPST). YCDUiPST cells are initialized in "DAPINIT", but "YAWDAP" is entered for the first time at a time corresponding to 1.5 complete DAP cycles (i.e. pitch and yaw, plus pitch), giving a derived rate for this first pass (at DAP turn-on or after a restart) that is 50% too big.

YCMD: Single precision value of previous yaw output command information, scale factor B14, units CDU actuator pulses (or B0 in units of ASCREV). See PCMD.

YDELOFF: Value of yaw c.g. offset, scale factor B14, units CDU actuator pulses (or B0 in units of ASCREV). Quantity is a double precision version of YACTOFF (cf. PDELOFF).

YERRB: Value of yaw error in body coordinates, loaded in "YCOPY" with ERRBTMP (which in turn is sum of previous YERRB and ($OMEGAC_z$ - $OMEGAB_z$)), scale factor B-1, units revolutions (cf. PERRB).

YTMPi (i = 1 - 6): Values of yaw filtered information, initialized to 0 in "TVCDAPON" and "SWICHOVR", and loaded with TMPi in "YCOPY". Scale factors are all B-1, units revolutions.

CM/POSE (Entered by setting AVEGEXIT = "CM/POSE" in "P62")

$mVREL\underline{L} = K_{mkvsc}\,\underline{V} + K_{kwe}\,(\underline{C}_{unitw} * UNIT\underline{R})$

$UX\underline{A} = \text{unit}(\,mVRE\underline{L})$

$T\underline{S} = -\,\text{unit}(\,UX\underline{A} * UNIT\underline{R})$

If $|mVRE\underline{L}|_{sp} - K_{spvq} < 0$:

$T\underline{S} = OLDUY\underline{A}$

$UY\underline{A} = T\underline{S}$

$OLDUY\underline{A} = UY\underline{A}$

$UZ\underline{A} = -\,UY\underline{A} * UX\underline{A}$

$T\underline{S}$ = (AOGdPIP, AIGdPIP, AMGdPIP), converted to ones complement double precision

$$UB\underline{Y} = (-\cos TS_x \cos TS_y \sin TS_z + \sin TS_x \sin TS_y,\ \cos TS_x \cos TS_z,\ \cos TS_x \sin TS_y \sin TS_z + \sin TS_x \cos TS_y)\ [REFSMMAT]$$

$$UB\underline{X} = (\cos TS_y \cos TS_z,\ \sin TS_z,\ -\sin TS_y \cos TS_z)\ [REFSMMAT]$$

$UB\underline{Z} = UB\underline{X} * UB\underline{Y}$

$T\underline{S}_1 = \text{unit}(\,UX\underline{A} * UB\underline{Y})$

$COSTH = T\underline{S}_1 \cdot UZ\underline{A}$

$SINTH = T\underline{S}_1 \cdot UY\underline{A}$

Perform "ARCTRIG"

$TS_x = THETA_{sp}$ (- roll angle)

$TS_y = (\sin^{-1}(UB\underline{Y} \cdot UX\underline{A}))_{sp}$ (- beta angle)

$SINTH = UB\underline{X} \cdot T\underline{S}_1$

$COSTH = T\underline{S}_1 \cdot UB\underline{Z}$

Perform "ARCTRIG"

$TS_z = THETA_{sp}$ (- alfa angle)

TS_2 = GAMA

$GAMA = -\left(\cos^{-1}(UNIT\underline{R} \cdot UZ\underline{A})\right)_{sp}$

Inhibit interrupts

If bit 11(GAMDIFSW) of FLAGWRD6 = 1:

    $GAMDOT = K_{tcdu} (GAMA - TS_2)$

    If $|GAMDOT| - K_{gmmn} < 0$:

        GAMDOT = 0

If bit 11(GAMDIFSW) of FLAGWRD6 = 0:

    Set bit 11(GAMDIFSW) of FLAGWRD6 = 1

    GAMDOT = 0

$TS_3 = - TS_x$, in range $\pm$ 180°

$ROLLdl80 = TS_3 - ROLLdPIP + ROLLdl80$, in range $\pm$ 180°

$TS_4 = - TS_z$, in range $\pm$ 180°

$ALFAdl80 = TS_4 - ALFAdPIP + ALFAdl80$, in range $\pm$ 180°

$BETAdl80 = - TS_y - BETAdPIP + BETAdl80$ (no overflow checks)

Release interrupts

$VMAGI = |\underline{V}|$

Proceed to address specified by POSEXIT

STARTENT (Entered by setting POSEXIT = "STARTENT" in "P63")

Set bits 9(RELVELSW), 8(EGSW), 7(NOSWITCH), 6(HIND), 5(INRLSW), and 3(05GSW) of FLAGWRD6 = 0

Set bits 13(ENTRYDSP), 10(GONEPAST), and 4(LATSW) of FLAGWRD6 = 1

$LOD = C_{odpad}$

$LAD = C_{adpad}$

$LdDCMINR = K_{cs15p2}$ LAD

$KLAT = K_{atspe}$ LAD

$Q7 = K_{q7f}$

$FACTOR = (1 - 2^{-28})$

LdD = - LAD sgn HEADSUP

Perform "STARTEN1"

LATANG = $\left(\text{unit}(\underline{V} * \text{UNIT}\underline{R})\right) \cdot \underline{R}_t$

K2ROLL = -1 sgn LATANG_{sp} (+0 is +, -0 is -)

Q2 = K_{q21} LAD + K_{q22}

GOTOADDR = "INITROLL"

POSEXIT = "SCALEPOP"

Proceed to "SERVEXIT"

STARTEN1 Entered from "STARTENT" and "NEWRNVN"

GOTOADDR = Return address

Set bit 13(ERADCOMP) of FLAGWRD1 = 0

Set bit 12(LUNLATLO) of FLAGWRD3 = 0

LAT = LATSPL

LONG = LNGSPL

ALT = DELVLVC_x

ALT = 0

TIMEdRTO = T_{pptm}

TS = TIMEdRTO

Perform "LALOTORV"

$\underline{R}_{ti}$ = unitALPHA$\underline{V}$

DTEAROT = K_{500sec}

Perform "EARROT1"

THETAH = $\cos^{-1}$ ($\underline{R}_t$ · UNIT$\underline{R}$)

Proceed to address specified by GOTOADDR

SCALEPOP Entered due to setting of POSEXIT in "STARTENT"

Perform "TARGETNG"

Proceed to address specified by GOTOADDR

TARGETNG

If bit 9(RELVELSW) of FLAGWRD6 = 0:

VEL$\underline{ }$ = K_{kvsc} $\underline{V}$

If bit 9(RELVELSW) of FLAGWRD6 = 1:

$\text{VE}\underline{\text{L}} = -\ \text{mVRE}\underline{\text{L}}$

$\text{UNIT}\underline{\text{V}} = \text{unitVE}\underline{\text{L}}$

$\text{VSQUARE} = |\text{VE}\underline{\text{L}}|^2$

$\text{LEQ} = \text{VSQUARE} - 1$

$\text{NV} = |\text{VE}\underline{\text{L}}|$

$\text{RDOT} = \text{VE}\underline{\text{L}} \cdot \text{UNIT}\underline{\text{R}}$

$D = K_{kasc}\ |\text{DEL}\underline{\text{V}}|$, limited $\geq K_{mnd}$ (limit effective only if $D_{tp} = 0$).

$\text{UN}\underline{\text{I}} = \text{unit}(\ \text{VE}\underline{\text{L}} * \text{UNIT}\underline{\text{R}})$

If bit 9(RELVELSW) of FLAGWRD6 = 1:

$\text{TS} = 0$

If bit 9(RELVELSW) of FLAGWRD6 = 0:

$D = D + D(\ \text{RDOT} / K_{mhsca} + K_{mksc}\ D / \text{NV}\ )$

If bit 8(EGSW) of FLAGWRD6 = 0:

$\text{TS} = K_{kteta}\ \text{THETAH}$

If bit 8(EGSW) of FLAGWRD6 = 1:

If $\text{NV} - K_{vmin} < 0$:

Set bit 9(RELVELSW) of FLAGWRD6 = 1

$\text{TS} = K_{kt1}\ \text{THETAH} / \text{NV}$

$\text{DTEAROT} = \text{TS} + T_{pptm} - \text{TIMEdRTO}$

Perform "EARROT2"

$\text{LATANG} = \underline{R}_t \cdot \text{UN}\underline{\text{I}}$

Set bit 8(GONEBYTG) of FLAGWRD7 = 0

If $(\underline{R}_t * \text{UNIT}\underline{\text{R}}) \cdot \text{UN}\underline{\text{I}} < 0$, set bit 8(GONEBYTG) of FLAGWRD7 = 1

$\text{TS} = \underline{R}_t \cdot \text{UNIT}\underline{\text{R}}$

If $\text{TS} - K_{nq} < 0$:

$\text{THETAH} = \cos^{-1} \text{TS}$

If $TS - K_{nq} \geq 0$:

$THETAH = K_{kacs} \sqrt{|2(TS - 1)|}$

TS = THETAH

If bit 8 (GONEBYTG) of FLAGWRD7 = 0:

TS = - TS

RTGON67 = TS

If $D - K_{pt05g} < 0$:

Set bit 3(05GSW) of FLAGWRD6 = 0

Return

Set bit 3(05GSW) of FLAGWRD6 = 1

$$TS = \frac{|\underline{DELVREF}|^2 - (\underline{DELVREF} \cdot \underline{UXA})^2}{(\underline{DELVREF} \cdot \underline{UXA})^2}$$

If no overflow has taken place (e.g. $|TS| < 1$):

$LdDCALC = \sqrt{TS}$

Return

INITROLL

If bit 5(INRLSW) of FLAGWRD6 = 1:

If $RDOT + K_{vrcont} \geq 0$:

DIFFOLD = 0

$DLEWD = K_{dlewd0}$

$LEWD = K_{ewdl}$

GOTOADDR = "HUNTEST"

Proceed to "HUNTEST"

If (KAT - D) < 0, proceed to "CONSTD"

Proceed to "LIMITL/D"

If bit 3(05GSW) of FLAGWRD6 = 0, proceed to "LIMITL/D"

TS = 64 and perform "NEWMODEX"

$ENTRYVN = 0674_{vn}$

$KAT = K_{ka2} + LEQ^3 / K_{ldkal}$, limited $\leq K_{kalim}$

Set bit 10(GONEPAST) of FLAGWRD6 = 0 ("STARTENT" initialized this bit to 1)

If NV $-$ K_{vfnl} < 0:

GOTOADDR = "KEP2"

Set bit 5(INRLSW) of FLAGWRD6 = 1 (must be 0 to get here)

Proceed to "LIMITL/D"

DO = K_{ka3} LEQ + K_{ka4} (NV at least 27,000 fps to get here)

CdDO = K_{cone} / DO

Reset overflow indicator

LdD = LAD

TS = $(\text{RDOT} / \text{NV})^3 / K_{ldk44} - K_{vfin} + \text{NV}$

If no overflow has taken place since indicator reset:

If TS ≥ 0:

LdD = - LAD

Set bit 5(INRLSW) of FLAGWRD6 = 1 (must be 0 to get here)

Proceed to "LIMITL/D"

HUNTEST

A1 = D

TS = LAD

If RDOT ≥ 0:

TS = LEWD

V1 = NV + RDOT / TS

AO = $(\text{V1}^2 / \text{VSQUARE}) \left(\text{D} + \text{RDOT}^2 / (K_{2chs}\ \text{TS})\right)$

If RDOT < 0:

A1 = AO

If LdD < 0:

V1 = V1 - K_{vquit}

Proceed to "HUNTEST1"

HUNTEST1

$ALP = K_{2chs}\ AO\ /\ (LEWD\ V1^2)$

$FACT1 = V1\ /\ (1 - ALP)$

$FACT2 = ALP\ (ALP - 1)\ /\ AO$

$VL = FACT1\ (1 - \sqrt{FACT2\ Q7 + ALP}\ )$

$GAMMAL1 = LEWD\ (V1 - VL)\ /\ VL$

If $VL - K_{vmn} < 0$:

Proceed to "PREFINAL"

$VBARS = VL^2$

$TS = K_{hav} - VL$

If $TS < 0$:

GOTOADDR = "HUNTEST"

Proceed to "CONSTD"

$DVL = TS$

$VS1 = K_{hav}$

$TS = VS1 - V1$

If $TS \geq 0$:

$DVL = DVL - TS$ (i.e. $V1 - VL$)

$VS1 = V1$

$$DHOOK = \frac{(1 - VS1\ /\ FACT1\ )^2 - ALP}{FACT2}$$

$AHOOKDV = \frac{1}{4}\ (DHOOK\ /\ Q7) - K_{chk}$

$TS = GAMMAL1 - (AHOOKDV + 1)\ K_{chone}\ DVL^2\ /\ (DHOOK\ VBARS)$

If $TS \geq 0$:

$GAMMAL = TS$

If TS $<$ 0:

$$VL = VL + \frac{(1/3)\ VL\ TS}{(1/3)\ LEWD\ -\ (AHOOKDV + 2/3)(K_{chone}\ DVL)/(DHOOK\ VL)}$$

$$Q7 = \frac{(1 - VL\ /\ FACT1\ )^2 - ALP}{FACT2}$$

$VBARS = VL^2$

GAMMAL = 0

$GAMMAL1 = GAMMAL1 + K_{q19}\ (GAMMAL - GAMMAL1)$

Proceed to "RANGER"

RANGER

$COSG = 1 - \frac{1}{2}\ GAMMAL^2$

$CPE = \sqrt{1 + (VBARS - 2)\ COSG^2\ VBARS}$

$ASKEP = 2\ \sin^{-1}\ (VBARS\ COSG\ GAMMAL\ /\ CPE\ )$

$ASP1 = Q2 + K_{q3}\ VL$

$ASPUP = (K_{c12}\ /\ GAMMAL1)\ \left(-\ \log_e\left((V1^2\ Q7)/(AO\ VBARS)\right)\right)$

$ASPDWN = K_{kc3}\ RDOT\ NV\ /(AO\ LAD)$

$ASP3 = K_{q5}\ (\ K_{q6} - GAMMAL)$

$ASPSpTM\underline{p} = \left(\ (ASKEP_{sp},\ ASP1_{sp})\ ,\ (ASPUP_{sp},\ ASPDWN_{sp}),\ ASP3\right)$

$DIFF = ASP3 + ASPDWN + ASPUP + ASP1 + ASKEP - THETAH$

If $|DIFF| - K_{25nm} < 0$:

TS = 65 and perform "NEWMODEX"

Establish "P65.1" (priority 13_8)

GOTOADDR = "UPCONTRL"

Proceed to address specified by GOTOADDR

If bit 6(HIND) of FLAGWRD6 = 0:

If DIFF $\geq$ 0: (note that polarity of DIFF the opposite of that in official equation documentation)

DIFFOLD = DIFF

$Q7 = K_{q7f}$

GOTOADDR = "HUNTEST"

Proceed to "CONSTD"

DLEWD = (DLEWD DIFF) / (DIFFOLD - DIFF)

If DLEWD + LEWD < 0:

DLEWD = $-\frac{1}{2}$ LEWD

Proceed to second previous line (recheck of sum with LEWD)

If overflow has taken place since calculation of DIFF:

LEWD = $(1 - 2^{-28})$

DIFFOLD = DIFF

Q7 = K_{q7f}

GOTOADDR = "HUNTEST"

Proceed to "CONSTD"

LEWD = DLEWD + LEWD

Change priority of present job to 16_8 (lower than "SERVICER")

GOTOADDR = "ENDEXIT"

DIFFOLD = DIFF

Set bit 6(HIND) of FLAGWRD6 = 1

Q7 = K_{q7f}

Proceed to "HUNTEST"

<u>P65.1</u> Established when P65 entered from "RANGER"

Set bit 13(ENTRYDSP) of FLAGWRD6 = 0

TS = 1669_{vn}

Perform "GOFLASHR": if terminate, proceed to previous line
if proceed, skip next line
otherwise, proceed to previous line

End of job

Set bit 13(ENTRYDSP) of FLAGWRD6 = 1

End of job

UPCONTRL

If $D - K_{c21} \geq 0$:

Set bit 7(NOSWITCH) of FLAGWRD6 = 1 (reset in "L355")

If $NV - V1 \geq 0$:

Reset overflow indicator

$TS = LAD + K_{k2d} \left(RDOT + LAD\ (NV - V1) \right)$ (scaled B8)

$TS_1 = AO\ VSQUARE\ /\ V1^2 - (V1 - NV)^2\ LAD\ /\ K_{2chs}$ (scale BO)

$TS_2 = K_{k1d}\ (D - TS_1) + TS$ (scaled B8)

Rescale TS_2 to scale factor BO

If overflow has taken place since indicator reset:

$LdD = LAD\ \mathrm{sgn}\ TS_2$

Proceed to "GLIMITER"

$LdD = TS_2$

Proceed to "NEGTESTS"

If $D - Q7 < 0$:

TS = 66 and perform "NEWMODEX"

$ENTRYVN = 0622_{vn}$

Set bit 13(ENTRYDSP) of FLAGWRD6 = 1

GOTOADDR = "KEP2"

Proceed to "KEP2"

If $RDOT < 0$:

If $NV - VL - K_{c18} < 0$:

Proceed to "PREFINAL"

If $D - AO \geq 0$:

LdD = LAD

Proceed to "LIMITL/D"

$VREF = FACT1 \left(1 - \sqrt{D\ FACT2 + ALP}\right)$

$RDOTREF = LEWD\ (V1 - VREF)$

$TS = VS1 - VREF$

If $TS \geqslant 0$:

$$RDOTREF = RDOTREF - \frac{K_{chone}\ TS^2\ (1 + TS\ AHOOKDV\ /\ DVL)}{DHOOK\ \ VREF}$$

$TS = D - K_{q7min}$ (Tag here "CONTINU2")

Reset overflow indicator

If $TS \geqslant 0$: (should not be, since $K_{q7min} = +MAX$)

$FACTOR = (D - Q7)\ /\ (A1 - Q7)$

$TS = FACTOR \left(NV - VREF + FACTOR\ (RDOT - RDOTREF)/K_{1dkb} \right) / K_{m1dkb2}$ (scale B4)

If overflow has taken place since indicator reset:

LdD = LAD sgn TS

Proceed to "GLIMITER"

If $|TS| - K_{ptb4} \geqslant 0$:

$TS = \left(K_{ptb4} + K_{ptone} \left(|TS| - K_{ptb4} \right) \right)$ sgn TS

Rescale TS to scale factor B0

$TS = TS + LEWD$

If overflow has taken place since indicator reset:

LdD = LAD sgn TS

Proceed to "GLIMITER"

LdD = TS

Proceed to "NEGTESTS"

NEGTESTS

If $D - K_{c20} < 0$:

Proceed to "LIMITL/D"

Set bit 4(LATSW) of FLAGWRD6 = 0

If LdD < 0:

LdD = 0

Proceed to "LIMITL/D"

CONSTD

Reset overflow indicator

$TS = CdDO\ LEQ + K_{k2d}\ (RDOT + K_{2hs}\ DO / NV)$ (scale B8)

$TS_2 = K_{kld}\ (D - DO) + TS$ (scaled B8)

Rescale TS_2 to scale factor B0

If overflow has taken place since indicator reset:

LdD = LAD sgn TS_2

Proceed to "GLIMITER"

LdD = TS_2

Proceed to "NEGTESTS"

KEP2

If $K_{q7fkdm} - D < 0$:

Proceed to "PREFINAL"

$TS_{tp} = (ROLLC_{dp}, ROLLHOLD)$

If bit 3(05GSW) of FLAGWRD6 = 0:

$TS_{tp} = 0$

$(ROLLC_{dp}, ROLLHOLD) = TS_{tp}$

Proceed to "P62.3" (ENTRYVN set to 0622_{vn} by "UPCONTRL" before "KEP2" entered, if in P66)

PREFINAL

GOTOADDR = "PREFINAL" (for restart considerations)

TS = 67 and perform "NEWMODEX"

$ENTRYVN = 0666_{vn}$

Set bit 13(ENTRYDSP) of FLAGWRD6 = 1

Set bit 8(EGSW) of FLAGWRD6 = 1

GOTOADDR = "PREDICT3"

Proceed to "PREDICT3"

PREDICT3

If $NV - K_{vquit} < 0$:

Establish "P67.1" (priority 16_8)

GOTOADDR = "P67.2"

Proceed to "P67.2"

JJ = 12

If $K_{vrfr_{JJ}} - NV_{sp} > 0$:

JJ = JJ - 1 and repeat check

$$GRAD = \frac{\left| K_{vrfr_{JJ}} - NV_{sp} \right|}{K_{vrfr_{JJ+1}} - K_{vrfr_{JJ}}}$$

i = JJ

JJ = JJ + 13

$F1V = K_{drda_i} + GRAD\ (K_{drda_{i+1}} - K_{drda_i})$

JJ = JJ + 13

$F2V = K_{drdd_i} + GRAD\ (K_{drdd_{i+1}} - K_{drdd_i})$

JJ = JJ + 13

$RDRFV = K_{rdtr_i} + GRAD\ (K_{rdtr_{i+1}} - K_{rdtr_i})$

JJ = JJ + 13

$RTOGOV = K_{rtgo_i} + GRAD\ (K_{rtgo_{i+1}} - K_{rtgo_i})$

JJ = JJ + 13

$DREFRV = K_{aref_i} + GRAD\ (K_{aref_{i+1}} - K_{aref_i})$

JJ = JJ + 13

$F3V = K_{drd_i} + GRAD\ (K_{drd_{i+1}} - K_{drd_i})$

Set TS = DREFRV and DREFRV = 0

$PREDANG = RTOGOV + F2V\ (RDRFV - RDOT) + F1V\ (D_{sp} + TS)$

$TS = PREDANG_{dp} - THETAH$

If bit 10(GONEPAST) of FLAGWRD6 = 1:

LdD = - LAD

Proceed to "GLIMITER"

If bit 8(GONEBYTG) of FLAGWRD7 = 1:

Set bit 10(GONEPAST) of FLAGWRD6 = 1

$DNRNGERR = K_{maxrng}$

LdD = - LAD

Proceed to "GLIMITER"

DNRNGERR = TS

Reset overflow indicator

TS = - 4 DNRNGERR / F3V

If overflow has taken place since indicator reset ($|TS| \geq 1$):

LdD = LAD sgn TS

Proceed to "GLIMITER"

TS = TS + LOD

If $|TS| \geq 1$:

LdD = LAD sgn TS

Proceed to "GLIMITER"

LdD = TS

Proceed to "GLIMITER"

GLIMITER

$TS = K_{gmxd2} - D$

If $TS \geq 0$:

Proceed to "LIMITL/D"

$TS = TS + K_{gmxd2}$

If $TS < 0$:

LdD = LAD

Proceed to "LIMITL/D"

$CPX = \sqrt{K_{2hs}\ TS\ (K_{ldgmx}\ LEQ + LAD) + K_{2hsgmxq}\ /\ VSQUARE}$

If $CPX + RDOT \geq 0$:

Proceed to "LIMITL/D"

LdD = LAD

Proceed to "LIMITL/D"

LIMITL/D

LdD1 = LdD

If bit 10(GONEPAST) of FLAGWRD6 = 1: (also set in "STARTENT" and reset in "INITROLL" after 0.05G sensed)

Proceed to "L355"

$CPY = KLAT\ VSQUARE + K_{atbias}$

If $|LdD| - LdDCMINR \geq 0$:

If LATANG sgn K2ROLL $<$ 0:

LdD1 = LdDCMINR sgn LdD

Proceed to "L355"

$CPY = \frac{1}{2}\ CPY$

If LATANG sgn K2ROLL $-$ CPY $<$ 0:

Proceed to "L355"

If bit 7(NOSWITCH) of FLAGWRD6 = 1: (set by "UPCONTRL")

Set bit 7(NOSWITCH) of FLAGWRD6 = 0

Proceed to "L355"

K2ROLL = - K2ROLL

Proceed to "L355"

L355

TS = LdD1 / LAD, with magnitude limited < 1 (due to interpreter divide order)

Set bit 7(NOSWITCH) of FLAGWRD6 = 0

ROLLC = $(\cos^{-1} TS)$ sgn K2ROLL

Proceed to "ENDEXIT"

ENDEXIT

If bit 13(ENTRYDSP) of FLAGWRD6 = 1:

TS = ENTRYVN

Perform "REGODSPR"

Proceed to "SERVEXIT" (after special check for new job)

P67.1 Established by "PREDICT3"

TS = 1667_{vn}

Proceed to "GOFLASH": if terminate, proceed
if proceed, proceed
otherwise, proceed to previous line

Set bits 2(CMDSTBY) and 1(GYMDIF) of FLAGWRD6 = 0

AVEGEXIT = "SERVEXIT"

Proceed to "GOTOPOOH"

P67.2

Set bit 13(ERADCOMP) of FLAGWRD1 = 0

ALPHA$\underline{V}$ = $\underline{R}$

Set bit 12(LUNLATLO) of FLAGWRD3 = 0

TS = T_{pptm}

Perform "LAT-LONG"

Proceed to "SERVEXIT"

Quantities in Computations

See also list of major variables and list of routines

A1: Value of drag (set to D or A0 in "HUNTEST") used to compute FACTOR, scale factor B0, G-units (25 g's).

AHOOKDV: Term in GAMMAL calculation (equal to AHOOK DVL), scale factor B4.

AIGdPIP: See General Program Control.

ALFAd180: Single precision angle, program notation "ALFA/180", giving the third Euler angle of the CM attitude (about UB$\underline{Y}$ in "pitch"), scale factor B-1, units revolutions.

ALFAdPIP: See General Program Control.

ALP: Quantity used to compute VREF in "UPCONTRL" (and computed itself in "HUNTEST1", where it is also used), scale factor B0.

ALPHA$\underline{V}$: See Coordinate Transformations.

ALT: See Display Computations: discussion concerning loading with $DELVLVC_x$ also applicable to "STARTEN1".

AMGdPIP: See General Program Control.

A0: Initial drag for "UPCONTRL", scale factor B0, G-units (25 g's).

AOGdPIP: See General Program Control.

ASKEP: Value of "Kepler range", scale factor B0, units revolutions, stored in push-down list location 0D. A single precision value is available in the ASKEP cell itself for generating telemetry data.

ASP1: Value of "final phase" range, scale factor B0, units revolutions, stored in push-down list location 2D. A single precision value is available in the ASP1 cell itself for generating telemetry data.

ASP3: Value of "gamma correction", scale factor B0, units revolutions.

ASPDWN: Value of "range down to pull-up", scale factor B0, units revolutions, stored in push-down list location 6D. A single precision value is available in the ASPDWN cell itself for generating telemetry data.

ASPSpTMp: "Vector" information loaded for telemetry purposes (see "RANGER" for format). Cells loaded are the same as those used for WBOD$\underline{Y}$ (RCS DAP) and OMEGA$\underline{C}$ (TVC DAP); program notation "ASPS(TM)".

ASPUP: Value of "up-range", scale factor BO, units revolutions, stored in push-down list location 4D. A single precision value is available in the ASPUP cell itself for generating telemetry data.

AVEGEXIT: See General Program Control.

BETAd180: Single precision angle, program notation "BETA/180", giving the second Euler angle of the CM attitude (about $UX\underline{A} * UB\underline{Y}$ in "yaw"), scale factor B-1, units revolutions.

BETAdPIP: See General Program Control.

C_{adpad}: Single precision erasable memory constant, program notation "LADPAD", scale factor BO, used to initialize LAD in "STARTENT".

C_{odpad}: Single precision erasable memory constant, program notation "LODPAD", scale factor BO, used to initialize LOD in "STARTENT".

$\underline{C}_{unitw}$: See General Program Control.

CdDO: Value of K_{cone}/ DO, computed in "INITROLL" and used in "CONSTD", scale factor B6.

COSG: Approximation to value of cos GAMMAL, scale factor B1.

COSTH: See Coordinate Transformations.

CPE: Value of eccentricity, scale factor B2 (stored in push-down list).

CPX: Value of intermediate quantity used in "GLIMITER", scale factor B1, V-units (see NV).

CPY: Value of lateral miss limit, scale factor B2, units of radians.

D: Value of total acceleration, scale factor BO, G-units (25 g's).

DELVLV$\underline{C}$: See Burn Control (and ALT).

DELVRE$\underline{F}$: See General Program Control.

DHOOK: Quantity used in computation of GAMMAL, scale factor BO.

DIFF: Difference between predicted and actual expected range, scale factor BO, units revolutions. Note that polarity of program definition is the reverse of that in official equation documentation.

DIFFOLD: Previous value of DIFF, scale factor BO, units revolutions.

DLEWD: Computed desired change in LEWD, scale factor BO.

DNRNGERR: Difference between predicted angle of travel and actual angle to target (i.e. PREDANG - THETAH) computed in "PREDICT3", scale factor B0, units revolutions, provided that bit 10(GONEPAST) of FLAGWRD6 = 0. If bit is set 1, DNRNGERR is set to K_{maxrng} (and not subsequently loaded) in P67.

D0: Value of "controlled constant D", scale factor B0, G-units (25 g's).

DREFRV: Value of K_{aref} table for present NV, same scaling as K_{aref}.

DTEAROT: See Coordinate Transformations.

DVL: Value of (VS1 - VL), scale factor B1, V-units (see NV).

ENTRYVN: Single precision cell loaded with verb-noun pattern (in proper format) for use in "ENDEXIT" for entry display.

F1V: Value of K_{drda} table for present NV, same scaling as K_{drda}.

F2V: Value of K_{drdd} table for present NV, same scaling as K_{drdd}.

F3V: Value of K_{drd} table for present NV, same scaling as K_{drd}.

FACT1: Value of quantity computed in "HUNTEST1", scale factor B1, V-units (see NV).

FACT2: Value of quantity computed in "HUNTEST1", scale factor B0.

FACTOR: Quantity used in "UPCONTRL" (initialized to 1 in "STARTENT"), scale factor B0.

GAMA: Single precision angle computed in "CM/POSE", scale factor B0, units revolutions, used to compute (by forming first difference with previous value) GAMDOT.

GAMDOT: Rate of change of GAMA per 0.1 second, scale factor B-1, units revolutions. Set to zero if magnitude is less than K_{gmmn}.

GAMMAL: Value of "flight path angle at VL", scale factor B0, units radians.

GAMMAL1: Value of "simple form of GAMMAL", scale factor B0, units radians. Stored in push-down list location 22D.

GOTOADDR: Single precision cell used to contain the starting address of the computations performed after "SCALEPOP" is performed, and used to control the phases of the entry computations. Prior to start of entry computations, used as storage for return address information.

GRAD: Single precision "gradient" of present velocity with respect to a table value, scale factor B0, used to permit linear interpolation in "PREDICT3".

HEADSUP: See Entry Preparation.

JJ: Single precision indexing parameter used in "PREDICT3", scale factor B14. It is incremented by 13 to select the next set of tabular information within an index loop (the incrementing is shown because the quantity is on the downlink).

K_{1dgmx}: Constant, program notation "1/GMAX", scale factor B-2, units of g^{-1}. Value is 0.5, corresponding to $(1/8) \times 2^2$, where first term is reciprocal of "GMAX" and second is scale factor.

K_{1dk44}: Constant, program notation "1/K44", scale factor B-1, units of (1/V-units). Stored value is 0.00260929464, corresponding to $(1/19749550) \times 25766.1973 \times 2^1$, where first term is reciprocal of "K44" (in units of fps), second is V-unit conversion, and third is scale factor.

K_{1dka1}: Constant, program notation "1/KA1", scale factor B6, units of G-units (25 g's) reciprocal. Value is 0.30048077, corresponding to $(1/1.3) \times 25 \times 2^{-6}$, where first term is reciprocal of "KA1" (in units of g's), second is G-unit conversion, and third is the scale factor.

K_{1dkb}: Constant, program notation "1/KB1", scale factor B0, value 0.400. Value corresponds to (1/2.5), or the reciprocal of the value of "KB1".

K_{2chs}: Constant, program notation "2C1HS", scale factor B2, value 0.0215983264. Value corresponds to $2 \times (1.25) \times 28500 \times 805 \times (25766.1973)^{-2} \times 2^{-2}$, where first term is equation factor of 2, second is value of C1, third is value of HS, fourth converts answer from fps^2 to G-units (of 25 g's @ 32.2 fps^2 = 805 fps^2), fifth converts velocity to fps from V-units, and sixth is scale factor.

K_{2hs}: Constant, program notation "2HS", scale factor B2, value 0.0172786611. Value corresponds to $2 \times 28500 \times 805 \times (25766.1973)^{-2} \times 2^{-2}$, where terms (except for the omission of the 1.25 factor for C1) have analogous significance to those for K_{2chs}.

$K_{2hsgmxq}$: Constant, program notation "2HSGMXSQ", scale factor B4, value 3.05717E-5. Value corresponds to the square of: $(2 \times 28500 \times 8 \times 32.2 \times (25766.1973)^{-2} \times 2^{-2})$, where first term is equation factor of 2, second is value of HS, third is value of GMAX (in g's), fourth converts GMAX to fps^2, fifth converts V-units to fps, and sixth is square root of scale factor.

K_{25nm}: Constant, program notation "25NM", scale factor B0, units revolutions. Value is 0.0011574074, corresponding to $25 \times (1/21600)$, where first term is "basic" value and second converts from nmi to revolutions (it is the reciprocal of $3437.7468 \times 2\pi$, or 60×360 for 60 nmi/degree).

K_{500sec}: Constant, program notation "500SEC", scale factor B28, units centi-seconds. Value is 50000×2^{-28}, corresponding to 500 seconds.

K_{aref}: Single precision set of constants, scale factor B0, G-units (of 25 g's). To convert to fps^2, multiply table entries by $32.2 \times 25 = 805$. Table starts at "VREFER +65". See below.

K_{atbias}: Constant, program notation "LATBIAS", scale factor B2, value 0.00003, units radians. Corresponds to 0.00012 radians, or (multiplying by 3437.7468) 0.412529616 nmi.

K_{atspe}: Constant, program notation "LATSLOPE", scale factor B-1, value 0.083333. Value corresponds to $(1/12) \times 2^{-1}$, or (1/24).

K_{c12}: Constant, program notation "C12", scale factor B-5, value 0.00684572901. Value corresponds to $28500 \times (1/21\ 202\ 900) \times (1/2\pi) \times 2^5$, where first term is HS, second is reciprocal of RE in feet (standard "entry radius"), third converts from radians to revolutions, and fourth is scale factor.

K_{c18}: Constant, program notation "C18", scale factor B1, V-units (see NV). Value is 0.0097026346, corresponding to $500 \times (1/25766.1973) \times 2^{-1}$, where first term is "basic" value (in fps), second converts to V-units, and third is scale factor.

K_{c20}: Constant, program notation "C20", scale factor B0, G-units (of 25 g's). Value is 0.26086957, corresponding to $210 \times (1/805)$, where first term is "basic" value (in fps^2), and second is conversion to G-units.

K_{c21}: Constant, program notation "C21", scale factor B0, G-units (of 25 g's). Value is 0.17391304, corresponding to $140 \times (1/805)$, where first term is "basic" value (in fps^2), and second is conversion to G-units.

K_{chk}: Constant, program notation "CHOOK", scale factor B4, value 2^{-6}, corresponding to 0.25×2^{-4}, where first term is "basic" value and second is scale factor.

K_{chone}: Constant, program notation "CH1", scale factor B-4, value 0.32×2^{1}. Value corresponds to $1.0 \times (1/25) \times 2^4$, where first term is the "basic" value, second converts to G-units, and third is scale factor.

K_{cone}: Constant, program notation "C001", scale factor B6, value -0.000625. Value corresponds to $(-1) \times (1/25) \times 2^{-6}$, where first term is "basic" value, second converts to G-units, and third is scale factor.

K_{cs15p2}: Constant, program notation "COS15", scale factor B0, value 0.965. Value corresponds to cosine 15.2 degrees.

K_{dlewd0}: Constant, program notation "DLEWDO", scale factor B0, value -0.05.

K_{drd}: Single precision set of constants, scale factor B-3, units revolutions. Multiply table entries by $21600 \times 2^{-3} = 2700$ to convert to nmi. Table starts at "VREFER +78". See below.

K_{drda}: Single precision set of constants, scale factor B-3, units revolutions/G-unit. Multiply table entries by 2700/805 to convert to units of nmi/fps^2. Table starts at "VREFER +13". See below.

K_{drdd}: Single precision set of constants, scale factor B-1, units revolutions/V-unit. Multiply table entries by 10800/25766.1973 to convert to units of nmi/fps ($10800 = \frac{1}{2}$ 21600). Table starts at "VREFER +26". See below.

K_{ewd1}: Constant, program notation "LEWD1", scale factor B0, value 0.15.

K_{gmmn}: Single precision constant, program notation "FIVE", scale factor B-1, units revolutions per 0.1 second. Value is 5×2^{-14}, corresponding to a rate of about 0.5493 degrees/second.

K_{gmxd2}: Constant, program notation "GMAX/2", scale factor B0, G-units. Value is 0.16, corresponding to $\frac{1}{2} \times 8 \times (1/25)$, where first term gives equation factor of $\frac{1}{2}$, second is GMAX (in g's), and third converts to G-units (of 25 g's).

K_{hav}: Constant, program notation "HALVE", scale factor B1, units V-units. Value is 0.5, corresponding to 25766.1973 fps (one V-unit), i.e. "VSAT".

K_{k1d}: Constant, program notation "K1D", scale factor B8, value 0.0314453125. Value corresponds to $0.01 \times 805 \times 2^{-8}$, where first term is "basic" value (corresponding to "C16"), second converts to fps^2, and third is scale factor.

K_{k2d}: Constant, program notation "K2D", scale factor B7, value -0.402596836. Value corresponds to $(-1) \times 0.002 \times 25766.1973 \times 2^{-7}$, where first term is equation factor, second is "basic" value ("C17"), third converts from V-units to fps, and fourth is scale factor.

K_{ka2}: Constant, program notation "KA2", scale factor B0, G-units. Value is 0.008, corresponding to $0.2 \times (1/25)$, where the first term is "basic" value (in g's) and second converts to G-units of 25 g's.

K_{ka3}: Constant, program notation "KA3", scale factor B-2, G-units. Value is 0.44720497, corresponding to $90 \times (1/805) \times 2^2$, where first term is "basic" value (in fps^2), second converts to G-units, and third is scale factor.

K_{ka4}: Constant, program notation "KA4", scale factor B0, G-units. Value is 0.049689441, corresponding to $40 \times (1/805)$, where first term is "basic" value (in fps^2), and second converts to G-units.

K_{kacs}: Constant, program notation "KACOS", scale factor B5, value 0.004973592, corresponding to $(1/2\pi) \times 2^{-5}$.

K_{kalim}: Constant, program notation "KALIM", scale factor B0, G-units. Value is 0.06, corresponding to $1.5 \times (1/25)$, where the first term is "basic" value (in g's) and second converts to G-units of 25 g's.

K_{kasc}: Constant, program notation "KASCALE", scale factor B-13, value 0.97657358. Value corresponds to $0.0585 \times (1/0.3048) \times (1/805) \times \frac{1}{2} \times 2^{13}$, where first term is accelerometer scale factor (meters/second per count), second converts to feet, third converts to G-units, fourth compensates for the two-second accelerometer sampling cycle, and fifth is scale factor.

K_{kc3}: Constant, program notation "KC3", scale factor B-2, value -0.0247622232. Value corresponds to $(-1) \times (1/21\ 202\ 900) \times (1/2\pi) \times (25766.1973)^2 \times (1/805) \times 2^2$, where first term is "equation" value, second is reciprocal of RE in feet ("entry radius"), third converts to revolutions, fourth converts V-units to fps, fifth converts to G-units, and sixth is scale factor.

K_{kt1}: Constant, program notation "KT1", scale factor B29, value $15.7788327 \times 2^{-14}$. Value corresponds to $(21\ 202\ 900) \times (1/25766.1973) \times 100 \times 2\pi \times 2^{-29}$, where first term is RE ("entry radius"), second converts V-units, third converts to centi-seconds, fourth converts THETAH to radians, and fifth is scale factor.

K_{kteta}: Constant, program notation "KTETA", scale factor B28, value $38.3495203 \times 2^{-14}$. Value corresponds to $10^3 \times 100 \times 2\pi \times 2^{-28}$, where first term is "basic" value, second converts to centi-seconds, third converts THETAH to radians, and fourth is scale factor.

K_{kvsc}: Constant, program notation "KVSCALE", scale factor B-6, value 0.81491944. Value corresponds to $(1/25766.1973) \times 100 \times (1/0.3048) \times 2^6$, where first term is one V-unit (in fps) conversion, second converts from centi-seconds to seconds, third converts from meters to feet, and fourth is scale factor.

K_{kwe}: Constant, program notation "KWE", scale factor B0, value $0.120056652 \times 2^{-1}$. Value corresponds to $1546.70168 \times (1/25766.1973)$, where first term is value in fps (corresponds to radius of 21 210 605 feet for a period of 86164.10 seconds) and second converts to V-units.

K_{maxrng}: Constant, program notation "MAXRNG", scale factor B0, units revolutions. Octal value is $16631_8\ 06755_8$, corresponding to about 0.46247666 revolutions. Value selected so that noun display of DNRNGERR for this information will be +9999.9 nmi (DSKY noun conversion factor is $2\pi \times 6{,}373{,}338$ meters = 21622.4965 nmi/rev).

K_{mldkb2}: Constant, program notation "-1/KB2", scale factor B-3, value -0.0077621078×2^4. Value corresponds to $(-1) \times (1/0.0025) \times (1/25766.1973) \times 2^3$, where first term is equation factor, second is reciprocal of basic constant ("KB2"), third converts V-units, and fourth is scale factor.

K_{mhsca}: Constant, program notation "-HSCALED", scale factor B1, value -0.55305018. Value corresponds to (-1) x 28500 x (1/25766.1973) x 2^{-1}, where first term is equation factor, second is value of HS, third converts V-units, and fourth is scale factor.

K_{mksc}: Constant, program notation "-KSCALE", scale factor B1, value -0.0312424837. Value corresponds to (-2) x 805 x (1/25766.1973) x 2^{-1}, where first term is "equation" value, second converts G-units to fps^2, third converts V-units to fps, and fourth is scale factor.

K_{mkvsc}: Constant, program notation "-KVSCALE", scale factor B-6, value -0.81491944, or - K_{kvsc}.

K_{mnd}: Constant, program notation "1BITDP", scale factor B0, G-units. Value is 2^{-28}, corresponding to about 10^{-7} g.

K_{nq}: Constant, program notation "NEAR1/4", scale factor B2, value $(\frac{1}{4} - 2^{-14})$. Value corresponds to $(1 - 2^{-12})$, or about 0.99976, i.e. cos $1\frac{1}{4}$°.

K_{pt05g}: Constant, program notation ".05G", scale factor B0, G-units. Value is 0.002, corresponding to 0.05 x (1/25), where first term is "basic" value (in g's), and second converts to G-units (of 25 g's).

K_{ptb4}: Constant, program notation "PT1/16", scale factor B4, value 0.1 x 2^{-4}, corresponding to a "basic" value of 0.1.

K_{ptone}: Constant, program notation "POINT1", scale factor B0, value 0.1.

K_{q3}: Constant, program notation "Q3", scale factor B-1, value 0.167003132. Value corresponds to 0.07 x 25766.1973 x (1/21600) x 2^1, where first term is "basic" value (in units of nmi/fps), second converts V-units, third converts to revolutions, and fourth is scale factor.

K_{q5}: Constant, program notation "Q5", scale factor B0, value 0.326388889. Value corresponds to 7050 x (1/21600), where first term is "basic" value (in units of nmi/rad, corresponding to 0.3 x 23500) and second converts from nmi to revolutions.

K_{q6}: Constant, program notation "Q6", scale factor B0, units radians. Value is 0.0349, or about 2°.

K_{q7f}: Constant, program notation "Q7F", scale factor B0, G-units. Value is 0.0074534161, corresponding to 6 x (1/805), where first term is "basic" value (in fps^2), and second converts to G-units.

K_{q7fkdm}: Constant, program notation "Q7FKDMIN", scale factor B0, G-units. Value is 0.0080745342, corresponding to 6.5 x (1/805), where first term is "basic" value (in fps^2), and second converts to G-units.

K_{q7min}: Constant, program notation "Q7MIN", scale factor B0, G-units. Constant is $(1 - 2^{-28})$, corresponding to about 25 g's (to disable the check in "UPCONTRL" to recompute FACTOR), or 805 fps^2.

K_{q19}: Constant, program notation "Q19", scale factor B0, value 0.5.

K_{q21}: Constant, program notation "Q21", scale factor B0, units revolutions. Value is 0.162037037, corresponding to 3500 x (1/21600), where first term is "basic" value (nmi), and second converts to revolutions.

K_{q22}: Constant, program notation "Q22", scale factor B0, units revolutions. Value is -0.092222222, corresponding to -1992 x (1/21600), where first term is "basic" value (nmi), and second converts to revolutions.

K_{rdtr}: Single precision set of constants, scale factor B-2, V-units. Multiply table entries by ($\frac{1}{4}$ x 25766.1973) to convert to fps. Table starts at "VREFER +39". See below.

K_{rtgo}: Single precision set of constants, scale factor B-3, units of revolutions. Multiply table entries by 2700 (i.e. 2^{-3} x 21600) to convert to nmi. Table starts at "VREFER +52". See below.

K_{spvq}: Single precision constant, program notation "SPVQUIT", scale factor B1, V-units. Octal value is 00476_8, corresponding to about 1000.2 fps.

K_{tcdu}: Single precision constant, program notation "TCDU", scale factor B-1, value 0.1. Value corresponds to $\frac{1}{2}$ x 0.1 x 2^1, where first term reduces angle rate from units of computing cycles (2 seconds) to seconds, second is "basic" value (to determine angle increment per 0.1 second), and third is scale factor.

K_{vfin}: Constant, program notation "VFINAL", scale factor B1, V-units. Value is 0.51618016, corresponding to 26600 x (1/25766.1973) x 2^{-1}, where first term is "basic" value (in fps), second converts to V-units, and third is scale factor.

K_{vfnl}: Constant, program notation "VFINAL1", scale factor B1, V-units. Value is 0.523942273, corresponding to 27000 x (1/25766.1973) x 2^{-1}, where first term is "basic" value (in fps), second converts to V-units, and third is scale factor.

K_{vmin}: Constant, program notation "VMIN", scale factor B1, V-units. Value is 0.25, corresponding to a velocity of $\frac{1}{2}$ (25766.1973) fps, or about 12883.1 fps.

K_{vmn}: Constant, program notation "V1MIN", scale factor B1, V-units. Value is 0.34929485, corresponding to 18000 x (1/25766.1973) x 2^{-1}, where first term is "basic" value (in fps), second converts to V-units, and third is scale factor.

K_{vquit}: Constant, program notation "VQUIT", scale factor B1, V-units. Value is 0.019405269, corresponding to 1000 x (1/25766.1973) x 2^{-1}, where first term is "basic" value (in fps), second converts to V-units, and third is scale factor.

K_{vrcont}: Constant, program notation "VRCONT", scale factor B1, V-units. Value is 0.0135836886, corresponding to 700 x (1/25766.1973) x 2^{-1}, where first term is "basic" value (in fps), second converts to V-units, and third is scale factor.

K_{vrfr}: Single precision set of constants, scale factor B1, V-units. Multiply table entries by (2 x 25766.1973) to convert to fps. Table starts at "VREFER". See below.

K2ROLL: Quantity, scale factor B0 (only the sign is used, with quantity initialized to a non-zero magnitude in "STARTENT"), used to determine sign of ROLLC (used for lateral control).

KAT: Quantity, scale factor B0, G-units, computed in "INITROLL" the first cycle that D is not less than K_{pt05g}, and used as the "drag to lift up if down".

KLAT: Value of lateral switch gain computed in "STARTENT", scale factor B0.

LAD: Value of "maximum L/D (minimum actual vehicle L/D)" initialized in "STARTENT" from C_{adpad}, scale factor B0.

LAT: See Coordinate Transformations.

LATANG: Value of "lateral range", scale factor B2, units radians. Program notation also "XRNGERR".

LATSPL: See Display Computations.

LdD: Value of "desired lift-to-drag ratio (vertical plane)", scale factor B0. Program notation "L/D".

LdD1: Value of LdD used to compute ROLLC (set to LdD at start of "LIMITL/D", and perhaps subsequently changed), scale factor B0.

LdDCALC: Value of L/D computed in "TARGETNG" (for telemetry purposes only), scale factor B0. Uses same cell as TTE (see Display Computations). Program notation "L/DCALC".

LdDCMINR: Value of LAD cos 15.2° computed in "STARTENT", scale factor B0. Program notation "L/DCMINR".

LEQ: Value of (VSQUARE - 1), scale factor B2, units of gravity (divide by 25 to convert to G-units).

LEWD: Value of "UPCONTRL" reference L/D, scale factor B0. Updated in "RANGER" and initialized in "INITROLL".

LNGSPL: See Display Computations.

LOD: Value of "Final Phase L/D" loaded in "STARTENT", scale factor BO, from C_{odpad}.

LONG: See Coordinate Transformations.

mVREL: Complement of relative velocity, program notation "-VREL", scale factor B1, V-units. Computed in "CM/POSE".

NV: Magnitude of VEL (velocity normalized by satellite velocity of 25766.1973 fps), scale factor B1, V-units. Program notation "V". One V-unit, i.e. 25766.1973 fps, corresponds to $\sqrt{\mu/RE}$, where RE = 21 202 900 (300 000' above 2.09029E7 feet).

OLDUYA: Previous value of UYA saved in "CM/POSE", scale factor B1.

POSEXIT: Single precision exit address from "CM/POSE". Set to "P62.3" at start of "P62"; set to "STARTENT" in "P63"; and set to "SCALEPOP" (the normal beginning of the entry computations) in "STARTENT".

PREDANG: Single precision value of "predicted range" computed in "PREDICT3", scale factor B-3, units revolutions (one revolution is 21600 nmi). Terms are double precision if possible.

Q2: Value of final-phase ranging information used in "RANGER" to compute ASP1, scale factor BO, units revolutions. It is initialized in "STARTENT" based on the specified value for LAD.

Q7: Value of minimum drag for "UPCONTRL", scale factor BO, G-units. Initialized in "STARTENT".

$\underline{R}_t$, $\underline{R}_{ti}$: See Coordinate Transformations.

RDOT: Value of altitude rate, scale factor B1, V-units.

RDOTREF: Value of reference RDOT computed in "UPCONTRL", scale factor B1, V-units.

RDRFV: Value of K_{rdtr} table for present NV, same scaling as K_{rdtr}.

ROLLC: Value of roll command output of equations, scale factor BO, units revolutions.

ROLLd180: Single precision angle, program notation "ROLL/180", giving the first Euler angle of the CM attitude (about UBX in "roll"), scale factor B-1, units revolutions.

ROLLdPIP: See General Program Control.

ROLLHOLD: See Digital Autopilot Entry Routines.

RTGON67: Value of THETAH (with negative sign if not yet past target) loaded in "TARGETNG" for use with nouns 64 and 67, scale factor BO, units revolutions. Program notation also "RTGON64".

RTOGOV: Value of K_{rtgo} table for present NV, same scaling as K_{rtgo}.

SINTH: See Coordinate Transformations.

THETA: See Coordinate Transformations.

THETAH: Value of "desired range" (angle between present position vector and appropriately rotated target vector), scale factor B0, units revolutions.

TIMEdRTO: Value of computer time for which $\underline{R}_{ti}$ is valid, scale factor B28, units centi-seconds. Loaded in "STARTEN1", with program notation "TIME/RTO".

UB$\underline{X}$, UB$\underline{Y}$, UB$\underline{Z}$: Unit vectors, scale factor B1, computed in "CM/POSE". They define the "body axis triad" in reference coordinates: if all three gimbal angles are zero, UB$\underline{X}$ = unit$\underline{X}$ [REFSMMAT] etc; otherwise, the sequence is about y,z,x axes (inner, middle, and outer respectively) to go from platform to body axes.

UN$\underline{I}$: Unit vector, scale factor B1, giving the "normal to the trajectory plane".

UNIT$\underline{V}$: Unit VE$\underline{L}$ vector, scale factor B1.

UX$\underline{A}$, UY$\underline{A}$, UZ$\underline{A}$: Unit vectors, scale factor B1, computed in "CM/POSE". They define the "trajectory triad" in reference coordinates. UX$\underline{A}$ is in the direction of mVRE$\underline{L}$; UY$\underline{A}$ is in the direction of $\underline{v} * \underline{r}$ (left constant if magnitude of mVRE$\underline{L}$ is less than K_{spvq}), and UZ$\underline{A}$ completes the right-handed coordinate system (in direction of UY$\underline{A}$ * $\underline{v}$). Note that UX$\underline{A}$ corresponds to "- UV$\underline{A}$" and UZ$\underline{A}$ corresponds to "UN$\underline{A}$".

V1: Value of initial velocity for "UPCONTRL", scale factor B1, V-units.

VBARS: Value of VL^2, scale factor B2 (since VL in units of VSAT, no explicit division by VSAT is necessary to obtain VBARS).

VE$\underline{L}$: Value of velocity used in entry computations (inertial or relative), scale factor B1, V-units.

VL: Value of exit velocity for "UPCONTRL", scale factor B1, V-units.

VMAGI: Magnitude of $\underline{V}$ computed at the end of "CM/POSE", scale factor B7, units meters/centi-second, computed for display purposes.

VREF: Value of reference velocity for "UPCONTRL", scale factor B1, V-units.

VS1: Value of the lesser of V1 or K_{hav} (V-unit factor), scale factor B1, V-units.

VSQUARE: Square of NV, scale factor B2(cf. discussion on VBARS).

Constants used in "PREDICT3" Reference
Table

Stored Values

i	K_{vrfr}	K_{drda}	K_{drdd}	K_{rdtr}	K_{rtgo}	K_{aref}	K_{drd}
			Multiply by 2^{-3}	Multiply by 2^{+3}			
0	.019288	-.010337	-.0478599	-.0134001	.00137037	-.051099	.004491
1	.040809	-.016550	-.0683663	-.013947	.003851852	-.074534	.008081
2	.076107	-.026935	-.1343468	-.013462	.00874074	-.101242	.016030
3	.122156	-.042039	-.2759846	-.011813	.017148	-.116646	.035815
4	.165546	-.058974	-.4731437	-.0095631	.027926	-.122360	.069422
5	.196012	-.070721	-.6472087	-.00806946	.037	-.127081	.104519
6	.271945	-.098538	-1.171693	-.006828	.063296	-.147453	.122
7	.309533	-.223611	-1.466382	-.00806946	.077889	-.155528	.172407
8	.356222	-.298148	-1.905171	-.0109791	.098815	-.149565	.252852
9	.404192	-.298148	-2.547990	-.0151496	.127519	-.118509	.363148
10	.448067	-.602557	-4.151220	-.0179817	.186963	-.034907	.512963
11	.456023	-.99999	-5.813617	-.0159061	.238148	-.007950	.558519
12	.67918	-.99999	-5.813617	-.0159061	.294185 185 (9 figures)	-.007950	.558519

Basic Values

i	K_{vrfr}	K_{drda}	K_{drdd}	K_{rdtr}	K_{rtgo}	K_{aref}	K_{drd}
0	993.9	-.03460	-.002507	-690.4	3.63	-41.12	12.19
1	2104.2	-.05548	-.003582	-718.7	10.38	-59.99	21.75
2	3922.1	-.09028	-.007035	-693.5	23.57	-81.51	43.34
3	6293.7	-.14105	-.014454	-608.6	46.31	-93.89	96.73
4	8530.0	-.19775	-.024790	-492.6	75.48	-98.56	187.37
5	10099.5	-.23726	-.033898	-416.0	99.87	-102.30	282.13
6	14015.4	-.33041	-.061399	-351.9	170.89	-118.71	329.43
7	15949.8	-.75007	-.076826	-416.0	210.28	-125.19	465.55
8	18355.9	-1.0002	-.099825	-565.8	266.80	-120.38	682.75
9	20828.1	-1.0002	-.13349	-780.8	344.26	-95.42	980.53
10	23089.6	-2.0209	-.21752	-926.7	504.77	-28.10	1384.94
11	23498.4	-3.3538	-.30459	-819.7	643.03	-6.39	1508.04
12	35000.8	-3.3538	-.30459	-819.7	794.31	-6.39	1508.04
	"VREF" fps	"DR/DA" nmi/fps^2	"DR/DRDOT" nmi/fps	"RDOTREF" fps	"RTOGO" nmi	"DREFR" fps^2	"DR/DL/D" nmi

The "Basic Values" were obtained by converting the octal memory information into the corresponding decimal values (using the appropriate scaling information), and therefore differ slightly from the results that would be obtained by multiplying the (nominal) "Stored Values" by the same scaling information.

Entry Preparation

P61

$EXTVBACT = 20000_8$ (lock out other extended verbs)

HEADSUP = -1

Perform "S61.1"

$TS = 0661_{vn}$

Perform "VNFLASHR": (if terminate, proceed to "GOTOPOOH")
if proceed, skip next line
otherwise, proceed to previous line

End of job

If HEADSUP > 0:

$ROLLC = K_{btl4}$

If HEADSUP = +0 or if HEADSUP < 0: (ROLLC not loaded for -0 HEADSUP)

$ROLLC = K_{rvsc}$ (|HEADSUP| -1), limited $\geq$ +0 (i.e. 0 if HEADSUP = -1)

$EXTVBACT = 20000_8$

Proceed to "NEWRNVN"

NEWRNVN

$MM = T_{pptm}$

Perform "STARTEN1"

Set bit 3(05GSW) of FLAGWRD6 = 0

$\underline{R}_{one} = \underline{R}$

$URON\underline{E} = unit\underline{R}_{one}$

$\underline{V}_{one} = \underline{V}$

$UN\underline{I} = unit\ (\underline{V}_{one} * URON\underline{E})$

If $(MM - T_{pptm}) < 0$:

Proceed to "NEWRNVN" (an Average-G update has occurred)

Perform "S61.2"

Perform "CLEARMRK" (resets EXTVBACT to 0)

TS = 0660_{vn}

Proceed to "VNFLASH": (if terminate, proceed to "GOTOPOOH")
if proceed, proceed
otherwise, proceed to previous line

TS = 1663_{vn}

Proceed to "GOFLASH": if terminate, proceed to "GOTOPOOH"
if proceed, skip next 2 lines
otherwise, proceed

EXTVBACT = 20000_8

Proceed to "NEWRNVN"

TS = 62 and perform "NEWMODEX"

DNLSTCOD = 1

Proceed to "P62"

P62

Perform "S61.1"

POSEXIT = "P62.3"

Inhibit interrupts (until released, routine is "CM/DAPIC")

TS = 8192 + 8192 + TIME1 - (least significant half of T_{pptm}), modulo 16384

(equivalent to $T_{now} - T_{pptm}$)

If $(TS - 5) > 0$:

TS = TS - 10 and repeat check(proceed to previous line)

TS = $|TS - 5|$, limited $\geq$ 1 centi-second

CMdGYMDT = TS

Call "READGYMB" in TS centi-seconds (5 cs out of phase with "READACCS")

Set bits 13(ENTRYDSP), 12(CMDAPARM), 11(GAMDIFSW), 3(05GSW), and 1(GYMDIF) of FLAGWRD6 = 0

Set bits 4(LATSW) and 2(CMDSTBY) of FLAGWRD6 = 1

BETAd180 = 0

SWdNDX = 1

Release interrupts

AVEGEXIT = "CM/POSE" (Tag here "P62.2")

TS = 00041_8

Perform "GOPERF1R": if terminate, proceed to "GOTOPOOH"
if proceed, skip next line
otherwise, proceed to previous line

End of job

Proceed to "CM/DAPON" (returns to "P62.1")

<u>P62.1</u>

TS = 0661_{vn}

Proceed to "GOFLASH": if terminate, proceed to previous line
if proceed, proceed
otherwise, proceed to previous line

If HEADSUP $>$ 0:

$ROLLC_{sp} = K_{btl4}$ (note that, contrary to "P61", low half not set)

If HEADSUP = +0 or if HEADSUP $<$ 0: (ROLLC not loaded if HEADSUP -0)

$ROLLC_{sp} = K_{rvsc}$ (|HEADSUP| - 1), limited $\geq$ +0 (i.e. 0 if HEADSUP = -1)

ALFACOM = $C_{alfapad}$

BETACOM = 0

P63FLAG = 1

ENTRYVN = 0622_{vn}

Set bit 13(ENTRYDSP) of FLAGWRD6 = 1

If bit 1 of CMDAPMOD = 1, End of job (CMDAPMOD = -0 or +1)

Proceed to "P63" (CMDAPMOD = +0; the -1 value should not exist at this point)

P62.3 (address set in "P62" into POSEXIT; entered from "KEP2")

$\underline{Y}_{dc}$ = $UY\underline{A}$ cos ROLLC + $UZ\underline{A}$ sin ROLLC

$\underline{X}_{dc}$ = ($\underline{Y}_{dc}$ * $UX\underline{A}$) sin $C_{alfapad}$ + $UX\underline{A}$ cos $C_{alfapad}$

$\underline{Z}_{dc}$ = $\underline{X}_{dc}$ * $\underline{Y}_{dc}$

$[X_{sm}]$ = [REFSMMAT]

Perform "CALCGA"

Proceed to "ENDEXIT"

WAKEP62 Called by "EXDAP"

Establish "P63" (priority 13_8)

End of task

P63 Established by "WAKEP62" or entered from "P62.1"

TS = 63 and perform "NEWMODEX"

POSEXIT = "STARTENT"

ENTRYVN = 0664_{vn}

P63FLAG = -1

Perform "CLEANDSP"

End of job

S61.1

60GENRET = Return address

Perform "R02BOTH"

If bit 1(AVEGFLAG) of FLAGWRD1 = 1: (true for entrance from "P62" if V37 not used to start P62)

Proceed to "S61.1A"

Perform "MIDTOAV2"

Call "S61.1C" in $TS+1_{sp}$ centi-seconds (TS set when exit from "MIDTOAV2")

End of job

S61.1C

Establish "S61.1A" (priority 13_8)

AVEGEXIT = "SERVEXIT"

Proceed to "PREREAD"

S61.1A

$\underline{TS} = \text{unit}\left([\text{REFSMMAT}] \; (\underline{V} * \underline{R}) \right)$

If $TS_y \geqslant 0$:

If $TS_y - K_{cs30im} \geqslant 0$, or if unit vector overflowed:

Proceed to address specified by 60GENRET

If $TS_y < 0$:

If $- TS_y - K_{cs30im} \geqslant 0$, or if unit vector overflowed:

Perform "ALARM" (pattern 1427_8) (IMU "reversed")

Skip next line

Perform "ALARM" (pattern 1426_8)

$TS = 0509_{vn}$

Perform "GODSPR"

Delay 10 seconds (by putting job to sleep via "DELAYJOB")

Proceed to address specified by 60GENRET

Quantities in Computations

See also list of major variables and list of routines

60GENRET: Single precision cell used to retain return address information from "S61.1".

ALFACOM: See Digital Autopilot Entry Routines.

AVEGEXIT: See General Program Control.

BETACOM: See Digital Autopilot Entry Routines.

BETAd180: See Entry Computations.

$C_{alfapad}$: Single precision erasable memory constant, program notation "ALFAPAD", scale factor B-1, units revolutions, giving the nominal trim angle (expected to be a negative number).

CMDAPMOD: See Digital Autopilot Entry Routines.

CMdGYMDT: See Digital Autopilot Entry Routines.

DNLSTCOD: See Telemetry.

ENTRYVN: See Entry Computations.

EXTVBACT: See Verb Definitions.

HEADSUP: Single precision quantity used to indicate desired initial attitude of vehicle: it is positive non-zero if a ROLLC of 180 degrees (lift down) is desired, which is a "heads up" attitude. Should be set to -1 (or 0) if a ROLLC of 0 degrees (heads down/ lift up) is desired. Scale factor is B14, and it is initialized at start of "P61" to -1 (before the N61 display which can be used to change it there).

K_{bt14}: Single precision constant, program notation "BIT14", scale factor B0, units revolutions. Value is 0.5, corresponding to 180 degrees. Used in "P61" as double precision information, with least significant half zero, for initializing ROLLC; in "P62.1", however, is single precision, with least significant half of ROLLC left alone.

K_{cs30im}: Constant, program notation "C(30)LIM", scale factor B1, value 0.566985. Used in program in such a way (a check on performing overflow) that effective value is 0.86603, or cosine 30 degrees: the value in memory is $1.0 - \frac{1}{2} \cos 30^{\circ}$.

K_{rvsc}: Dummy constant used to indicate rescaling of HEADSUP information from a scale factor of B14 to a scale factor of B0 revolutions, "value" 2^{-14} (see HEADSUP). If HEADSUP were loaded in N61 with a value of -04097, for example, then ROLLC would be set to $+90^{\circ}$.

MM: Storage for state vector time in "NEWRNVN", to ensure a consistent state vector for "S61.2", scale factor B28, units centi-seconds. It is the time tag for $\underline{R}_{one}$ and $\underline{V}_{one}$, and hence is used in "S61.2" to compute TTE2 (see Display Computations).

P63FLAG: See Digital Autopilot Entry Routines.

POSEXIT: See Entry Computations.

$\underline{R}_{one}$: See Display Computations.

ROLLC: See Entry Computations.

SWdNDX: See Digital Autopilot Entry Routines.

UN$\underline{I}$: See Entry Computations (computed in "NEWRNVN" for "S61.2").

URONE: See Display Computations.

UX$\underline{A}$, UY$\underline{A}$, UZ$\underline{A}$: See Entry Computations.

$\underline{V}_{one}$: See Display Computations.

General Program Control

<u>PREREAD</u> Entered from "TIGAVEG", "TIGON", and "S61.1C"

Establish "LASTBIAS" (priority 21_8)

Perform "PREREAD1"

Establish "NORMLIZE" (priority 32_8)

Call "READACCS" in 2 seconds

End of task

<u>PREREAD1</u> Entered from "PREREAD" and "P11"

Perform "PIPASR"

Set bit 1(AVEGFLAG) of FLAGWRD1 = 1 (means Average-G not to be stopped)

Set bit 15(DRIFTFLG) of FLAGWRD2 = 0

Set bit 6(V37FLAG) of FLAGWRD7 = 1 (means Average-G running)

DVTOTAL = 0

Return

<u>NORMLIZE</u> Established by "PREREAD" and entered from "P11"

Inhibit interrupts

$\underline{R} = \underline{R}_{n1}$

$\underline{V} = \underline{V}_{n1}$

$T_{pptm} = T_{pptm1}$

Release interrupts

$T\underline{S} = \underline{R}$

Perform "CALCGRAV"

$GD\underline{T} = GDT\underline{1}$

$GOB\underline{L} = GOBL\underline{1}$

End of job

READACCS

Perform "PIPASR"

PIPCTR = 2

If bit 2(CMDSTBY) of FLAGWRD6 = 1:

AOGdPIP = AOG

AIGdPIP = AIG ("least significant halves" of AOGdPIP and AOG respectively, since a double precision order is used)

AMGdPIP = AMG

ROLLdPIP = ROLLd180

ALFAdPIP = ALFAd180 ("least significant halves" of ROLL information)

BETAdPIP = BETAd180

If bit 12(CMDAPARM) of FLAGWRD6 = 1:

Call "QUIKREAD" in 0.5 second

XOLDBU$\underline{F}$ = XPIPBU$\underline{F}$

XPIPBU$\underline{F}$ = $DEL\underline{V}_{sp}$ (loads uncompensated PIPA data as read)

CMdGYMDT = 5

Call "SETJTAG" in 1.2 seconds

If bit 1(AVEGFLAG) of FLAGWRD1 = 0: (Average-G to be stopped) (set 0 by "ISITPOO")

AVEGEXIT = "AVGEND"

If bit 1(AVEGFLAG) of FLAGWRD1 = 1:

Call "READACCS" in 2 seconds

Establish "SERVICER" (priority 20_8)

Set bit 9(Test connector Outbit) of channel 11 = 1

End of task

SERVEXIT

End of job (after resetting restart group 5 for accelerometer reading)

QUIKREAD (Called by "READACCS" if entry DAP is on)

XOLDBUF = XPIPBUF

XPIPBUF = PIPA

If PIPCTR > 0:

PIPCTR = PIPCTR - 1

Delay 0.5 seconds

Proceed to "QUIKREAD"

End of task

AVGEND

1dPIPADT = (least significant half of T_{pptm})

Set bit 15(DRIFTFLG) of FLAGWRD2 = 1

Perform "AVETOMID"

Perform "PIPFREE"

MRKBUF1 = -256 (i.e. negative non-zero, keeping "REND1" from using)

Set bit 9(Test connector Outbit) of channel 11 = 0

Set bit 2(CMDSTBY) of FLAGWRD6 = 0

Set bit 6(V37FLAG) of FLAGWRD7 = 0 (Average-G not running)

If bit 9(UTFLAG) of FLAGWRD8 = 1:

Set restart group 1 to phase 11 (i.e. 1.11, causing "PIKUP20" to be established with priority 10_8 via restart logic)

Set restart group 2 to phase 0 (i.e. no effect, since inactive)

If bit 7(RNDVZFLG) of FLAGWRD0 = 1: (and UTFLAG = 0)

Set restart group 1 to phase 11 (i.e. 1.11, see above)

Set restart group 2 to phase 13 (i.e. 2.13, causing "REDOR22" to be established with priority 10_8 via restart logic)

Proceed to "CANV37"

SERVXT1

If bits 12-7 (Translation hand controller complement) of channel 31 $\neq 77_8$:

Set bit 10(BURNFLAG) of FLGWRD10 = 1 (Note that check only done once every 2 seconds, not once every 0.1 second in RCS DAP).

Proceed to "SERVEXIT"

SERVICER

If one or more components of $\underline{DELV}_{sp}$ has magnitude above K_{mxdlv}:

Perform "ALARM" (pattern 0205_8)

Proceed to "AVERAGEG"

Perform "1/PIPA"
If bit 3(05GSW) of FLAGWRD6 = 0:

$TTE = TTE2 + T_{pptm1}$

$DVTOTAL = DVTOTAL + K_{pp1} \left| \underline{DELV} \right|$ (could be incremented twice if get a restart)

Proceed to "AVERAGEG"

AVERAGEG

Perform "CALCRVG"

Inhibit interrupts

$\underline{R} = \underline{R}_{n1}$

$\underline{V} = \underline{V}_{n1}$

$T_{pptm} = T_{pptm1}$

$GD\underline{T} = GDT\underline{1}$

$GOB\underline{L} = GOBL\underline{1}$

Release interrupts

Proceed to address specified by AVEGEXIT

CALCRVG

$DELVRE\underline{F} = K_{pp1}\ DEL\underline{V}\ [REFSMMAT]$

$\underline{R}_{n1} = \underline{R} + K_{2s22}\ (\underline{V} + \frac{1}{2}\ DELVRE\underline{F} + \frac{1}{2}\ GD\underline{T})$

$T\underline{S} = \underline{R}_{n1}$

Perform "CALCGRAV"

$\underline{V}_{n1} = \underline{V} + DELVRE\underline{F} + \frac{1}{2}\ GD\underline{T} + \frac{1}{2}\ GDT\underline{1}$

Return

CALCGRAV

$\underline{UNITR} = unit\underline{TS}$

$X1 = - RTX2$ (loads with -0 for earth, -2 for moon)

$\underline{TS}_1 = \underline{UNITR}$

If $X1 \geqslant 0$: (i.e. earth)

$SIN\varnothing = \underline{TS}_1 \cdot \underline{C}_{unitw}$

$$\underline{GOBL1} = K_{20j} \frac{K_{resq}}{|\underline{TS}|^2} (0.05 - \tfrac{1}{4} SIN\varnothing^2)\ \underline{UNITR} \quad +$$

$$K_{2j} \frac{K_{resq}}{|\underline{TS}|^2}\ SIN\varnothing\ \underline{C}_{unitw}$$

$\underline{TS}_1 = \underline{GOBL1} + \underline{UNITR}$

$\underline{GDT1} = (K_{mmudt_{-X1}} / |\underline{TS}|^2)\ \underline{TS}_1$ (denominator formed triple precision, then normalized and used double precision for division, after which answer shifted right before $\underline{TS}_1$ multiplication)

Return

T4RUPT Entered based upon program interrupt #4, controlled by TIME4

If DSRUPTSW < 0:

If bit 14 of DSRUPTSW = 1:

If NOUT = 0, proceed to "NODSPY"

Perform "DSPOUTSB": if return to calling address +1, proceed to "NODSPY" otherwise, proceed

Set bit 14 of DSRUPTSW = 0

Set TIME4 to cause program interrupt #4 in 20 ms

$DSRUPTSW - DSRUPTSW + K_{1b6}$

Resume

OUTO = 0

Set bit 14 of DSRUPTSW = 1

(If DSRUPTSW $<$ 0):

Set TIME4 to cause program interrupt #4 in 20 ms

DSRUPTSW = DSRUPTSW + K_{1b6}

Resume

TS = DSRUPTSW

If TS $>$ 0:

DSRUPTSW = DSRUPTSW - 1, limited $\geq$ +0

If $|TS|$ = 0:

DSRUPTSW = 3

TS_1 = DSRUPTSW (program tag for TS_1 is RUPTREG1)

If DSPTAB+11 is flagged for output (i.e. is negative): (Tag "CDRVE")

Set TS = DSPTAB+11 and DSPTAB+11 = |DSPTAB+11| - 1

TS = (bits 11-1 of TS) (note if get a restart before DSPTAB+11 reloaded, "GOPROG" could put IMU into coarse align)

DSPTAB+11 = TS

OUTO = TS + $K_{rtu_{11}}$

Proceed to "HANG20"

If bit 15(DSKYFLAG) of FLAGWRD5 = 0: (Tag here "DSPOUT")

Proceed to "NODSPOUT"

If NOUT = 0:

Proceed to "NODSPOUT"

Perform "DSPOUTSB": if return to calling address +1, proceed to "NODSPOUT"
otherwise, proceed

Proceed to "HANG20"

<u>NODSPY</u>

OUTO = 0

i = 0

i = i + 1

DSRUPTSW = DSRUPTSW + K_{1b6}

If DSRUPTSW < 0:

Proceed to 3rd line of "NODSPY"

Set TIME4 to cause program interrupt #4 in (20 i) ms

Resume

DSPOUTSB

NOUT = NOUT - 1

TS_2 = -0 (program tag for TS_2 is DSRUPTEM)

If $DSPTAB_{DSPCNT}$ flagged for output (is negative):

Remove output flag from $DSPTAB_{DSPCNT}$ (form absolute value)

OUTO = (bits 15-11 of $K_{rtu_{DSPCNT}}$) + (bits 11-1 of $DSPTAB_{DSPCNT}$)

Return to calling address +2 (DSKY display sent)

If DSPCNT > 0:

DSPCNT = DSPCNT - 1

Proceed to third line of "DSPOUTSB"

If TS_2 = -0:

TS_2 = +0

DSPCNT = 10

Proceed to third line of "DSPOUTSB"

NOUT = 0

Return to calling address +1 (No DSKY display sent)

HANG20

DSRUPTSW = DSRUPTSW - 22400_8 (subtracts 5 scaled B6, and also forces bit 14 to be 0)

Set TIME4 to cause program interrupt #4 in 20 ms

Proceed to "PROCEEDE"

NODSPOUT

OUTO = 0

Set TIME4 to cause program interrupt #4 in 120 ms

Proceed to "PROCEEDE"

PROCEEDE

TS = channel 32 (bit 14 is PRO key complement)

If bit 14 of TS ≠ bit 14(PROCDBIT) of IMODES33:

Set bit 14(PROCDBIT) of IMODES33 = bit 14 of TS

If bit 14(PROCDBIT) of IMODES33 = 0: (i.e. 1 to 0 binary transition)

Establish "PROCKEY" (priority 30_8)

If TS_1 = 0, proceed to "OPTTEST"

If TS_1 = 1, proceed to "OPTMON"

If TS_1 = 2, proceed to "OPTTEST"

If TS_1 = 3, proceed to "IMUMON"

(TS_1, program notation RUPTREG1, has DSRUPTSW from "T4RUPT" before any modification by "HANG20")

SLAP1 (Entered via V36E)

Inhibit interrupts

Perform "STARTSUB"

DSPTAB+11 = bits 6(Gimbal lock) and 4(No Attitude) of DSPTAB+11, and flag for output at next opportunity

ERCOUNT = 0

FAILREG+i = 0 (i = 0 - 2)

REDOCTR = 0

Channel 77 = 0 (i.e. reset restart monitor flip-flops)

DSRUPTSW = - K_{12b5}

Proceed to "DOFSTART"

DOFSTART

ERESTORE = 0

SMODE = 0

UPSVFLAG = 0

Channel 5 = 0

Channel 6 = 0

Channel 11 = 0

Channel 12 = 0

Channel 13 = 0

Channel 14 = 0

WTOPTION = 0

DNLSTCOD = 0

NVWORD+2 = 0

DSPFLG+2 = 0

TRKMKCNT = 0 (Note these cells also used for RATETEMP, employed

VHFCNT = 0 in P23 and P24, so this initialization not effective)

EXTVBACT = 0

TRATE = 0

SRATE = 0

TOLD = 0

SOLD = 0

If bits 6 and 4 of DSPTAB+11 = 11_2: (coarse align w/gimbal lock)

Set bits 6(Enable IMU CDU Error Counters) and 4(Coarse Align) of channel 12 = 1

Make all restart groups inactive (by setting PHASEi = +0 and -PHASEi = -0 for i = 1 - 6)

MODREG = -0 (see "DSPMMJB")

RESTREG = 30000_8 (i.e. priority 30_8)

IMODES30 = 37411_8 (bits 14, 13, 12, 11, 10, 9, 4, and 1 set 1)

OPTIND = -1

OPTMODES = 00130_8 (bits 7, 5, and 4 to 1)

IMODES33 = 16000_8 (bits 13, 12, and 11 to 1)

T5LOC = "T5IDLOC"

FLAGWRDO = 0

FLAGWRD1 = bit 12(NODOP01) of FLAGWRD1 (other bits set 0)

FLAGWRD2 = 0

FLAGWRD3 = bit 13(REFSMFLG) of FLAGWRD3 (other bits set 0)

FLAGWRD4 = 0

FLAGWRD5 = 00200_8 (bit 8(CMCCOMP) is set 1)

FLAGWRD6 = 00004_8 (bit 3(05GSW) is set 1)

FLAGWRD7 = 0

FLAGWRD8 = bits 12(CMOONFLG), 11(LMOONFLG) and 8(SURFFLAG) of FLAGWRD8 (other bits set 0)

FLAGWRD9 = 0

FLGWRD10 = bit 11(HDSUPFLG) of FLGWRD10 (other bits set 0)

FLGWRD11 = 0

Proceed to second line of "DUMMYJOB"

<u>GOPROG</u> (Entered based on program interrupt #11, for a hardware restart (interrupts inhibited before entrance)

REDOCTR = REDOCTR + 1

TS = BBANK + SUPERBNK (or'ed into bits 7-5) (BBANK is that when restart generated)

RSBBQ = (TS, QREG)

Perform "VAC5STOR"

TS = Channel 33

Reset flip-flop bits of channel 33 (bits 15-11, done by write-type order)

If bit 15(Oscillator stop complement) of TS = 1:

    If bit 14(Computer warning complement) of channel 33 = 0:

        Perform "STARTSUB"

        Proceed to "DOFSTART" (conclude in restart loop due e.g. to memory parity failure)

Perform "LIGHTSET" (which may exit to "DOFSTART")

If bits 15-11 of ERESTORE $\neq$ 0: (Note ERESTORE cell is <u>not</u> excluded from normal memory test sequence in "ERASLOOP")

Perform "STARTSUB"

Proceed to "DOFSTART"

If ERESTORE $\neq$ 0:

If ERESTORE $\neq$ SKEEP7:

Perform "STARTSUB"

Proceed to "DOFSTART"

EBANK = (bits 11-9 of SKEEP4)

$E_{SKEEP7} = SKEEP5_{dp}$

ERESTORE = 0

Perform "STARTSUB"

TS = bits 15-14(DAPBIT1, DAPBIT2) of FLAGWRD6 (Tag here "ELRSKIP")

Set T5LOC according to the following table:

TS	T5LOC
00_2	"T5IDLOC"
01_2	"REDORCS"
10_2	"REDOTVC"
11_2	"REDOSAT"

Set bit 14(INTINUSE) of FLGWRD10 = 0

TS = (bits 5-4 of OPTMODES)

OPTMODES = TS + 00100_8 (bit 7, OCDUFBIT, set 1)

TS = bit 6(NOIMUDAP) of IMODES33

IMODES33 = TS + 16000_8 (bits 13-11 to 1)

TOLD = 0

SOLD = 0

DSPTAB+11 = bits 9(Program alarm), 6(Gimbal lock), and 4(No Attitude) of DPSTAB+11, and flag for output at next opportunity

If bit 4(No Attitude) of DSPTAB+11 = 1:

Perform "SETCOARS"

Call "CA+ECE" in 0.06 seconds

TS = (bits 9, 5, 4, 3, and 1 of IMODES30) (Tag here "NOCOARSE")

IMODES30 = TS + 37000_8 (sets bits 14-10 to 1)

If bit 7(ENGONFLG) of FLAGWRD5 = 1:

Set bit 13(SPS Engine On) of channel 11 = 1

Proceed to "GOPROG3"

<u>ENEMA</u> (Entered from "BAILOUT", "POODOO", "TRACKTRM", and "VERB94")

Inhibit interrupts

Perform "LIGHTSET" (which may exit to "DOFSTART")

Perform "STARTSB2"

Set bit 14(INTINUSE) of FLGWRD10 = 0

If bits 15-14(DAPBIT1, DAPBIT2) of FLAGWRD6 = 10_2: (TVC DAP)

Call "TVCEXEC" in 0.5 seconds

Proceed to "GOPROG3"

<u>GOPROG2</u> (Entered from "REV37" and "R00")

Inhibit interrupts

Perform "STARTSB2"

Set bit 14(INTINUSE) of FLGWRD10 = 0

Proceed to "GOPROG4"

<u>GOPROG3</u>

If bits 15-14(DAPBIT1, DAPBIT2) of FLAGWRD6 = 01_2: (RCS DAP)

If bit 5(TRACKFLG) of FLAGWRD1 = 0:

Perform "STOPRATE"

Proceed to "GOPROG4"

<u>GOPROG4</u>

If PHASEi $\neq$ (-1) (-PHASEi): (i = 1 - 6)

Perform "ALARM" (pattern 1107_8)

Proceed to "DOFSTART"

Establish "DSPMMJB" (priority 30_8)

Establish appropriate jobs and call appropriate tasks based on information in the active restart groups

If all PHASEi = 0: (i = 1 - 6, no active restart groups)

If bit 15 of MODREG = 0 (i.e. cell not -0)

Proceed to "GOTOPOOH" (Note "VNFLASH" will put the job in core set 0, i.e. "DSPMMJB", to sleep, leaving program number display blank)

Proceed to second line of "DUMMYJOB"

<u>LIGHTSET</u> (Entered from "GOPROG" and "ENEMA")

If bit 7(Mark reject) of channel 16 = 1:

If bits 5-1 of channel 16 = 22_8: (Error reset)

Perform "STARTSUB"

Proceed to "DOFSTART"

If channel 15 = 22_8: (Error reset from main panel DSKY)

Perform "STARTSUB"

Proceed to "DOFSTART"

Return

<u>STARTSUB</u>

DNTMGOTO = "DNPHASE1"

Set TIME5 to cause program interrupt #5 in 40 ms

Set TIME4 to cause program interrupt #4 in 30 ms

Proceed to "STARTSB2"

<u>STARTSB2</u>

Set bits 7-3 (Operator Error, Flash, Key Release, Temperature Caution, and Uplink Activity) of channel 11 = 0 (other bits left alone)

Set TIME3 to cause program interrupt #3 in 10 ms

Set bits 11-10 (Enable Standby, Test DSKY lights) of channel 13 = 0
(other bits left alone)

Set bits 14(R21MARK) and 12(P21FLAG) of FLAGWRD2 = 0

Set bit 10(SKIPVHF) of FLAGWRD2 = 1 (flag to "VHFREAD" that got restart)

Set bit 4(MARKFLG) of FLAGWRD1 = 0

Set bit 4(CYC61FLG) of FLAGWRD0 = 0

Set waitlist times to 81.93 seconds

Set waitlist task starting addresses to "SVCT3"

PRIORITY_i = -0 (i = 0 - 6)

DSRUPTSW = -0

NEWJOB = -0

Make all VAC areas available (by setting VAC<u>I</u>USE = "VAC<u>I</u>USE"
for <u>I</u> = 1 - 5)

DSPTAB+i = 2^{11} and flag for output at next opportunity (i = 0 - 10)
(blanks DSKY displays)

DELAYLOC+i = 0 (i = 0 - 3)

INLINK = 0

DSPCNT = 0

CADRSTOR = 0

REQRET = 0

CLPASS = 0

DSPLOCK = 0

MONSAVE = 0

MONSAVE1 = 0

VERBREG = 0

NOUNREG = 0

DSPLIST = 0

IMUCADR = 0

LGYRO = 0

FLAGWRD4 = 0

MARKINDX = 0

EXTVBACT = 0

NOUT = 11

SELFRET = "SELFCHK"

DSPCOUNT = -19

Return

GOTOPOOH

Set bit 7(AUTOSEQ) of FLGWRD10 = 0

Set restart group 4 to cause a start at next line (tag here "MNKGOPOO")

Perform "INITSUB"

Set bit 1(XDSPFLAG) of FLAGWRD4 = 0

Perform "AUTOCHK"

TS = 3799_{vn} (noun not permanently displayed, but must be non-zero)

Proceed to "VNFLASH": (if terminate, proceed to "GOTOPOOH")
if proceed, proceed to previous line
otherwise, proceed to previous line

(An input of XXE will cause "V37" entrance from "MMCHANG")

V37 (Entered from "MMCHANG" or "VERB96" with TS = new program number)

MMNUMBER = TS

Set bit 7(AUTOSEQ) of FLGWRD10 = 0

Set bit 15(PCMANFLG) of FLGWRD10 = 0

Set bit 9(VHFRFLAG) of FLAGWRD9 = 0

MRKBUF1 = -1

If bit 13(REFSMFLG) of FLAGWRD3 = 0:

Proceed to "AUTO37" (note that P81-P86 not rejected, and P79 accepted at this point without P20)

If MMNUMBER = 79:

TEMPMM = MMNUMBER

Proceed to second line of "REND30S"

If (MMNUMBER - 79) $>$ 0:

Set bit 7(Operator error) of channel 11 = 1

Perform "RELDSP"

Proceed to "PINBRNCH"

If (MMNUMBER - 30) $\leq$ 0:

Proceed to "AUTO37"

If (MMNUMBER - 36) $\leq$ 0:

Proceed to "REND30S"

(Note that should not do this with Average-G running and RNDVZFLG = 0, since do P20 startup and since TEMPMM also used for least significant half of $GDT1_z$)

Proceed to "AUTO37"

<u>AUTO37</u> (Entered from "AUTOSET", "P85", "REND30S", and "V37")

RESTREG = 30000_8 (i.e. priority 30_8)

TOLD = 0

SOLD = 0

TRATE = 0

SRATE = 0

Set bit 14(P24FLAG) of FLAGWRD9 = 0

If bit 6(IMUNITBT) of IMODES30 = 1:

Perform "ALARM" (pattern 1520_8)

Perform "RELDSP"

Proceed to "PINBRNCH"

If bit 7(ENGONFLG) of FLAGWRD5 = 0:

If bits 15-14(DAPBIT1, DAPBIT2) of FLAGWRD6 $\neq 10_2$:

Proceed to "ISITP00" (not TVC DAP)

Perform "SPSOFF" (exits with interrupts inhibited)

Perform "MASSPROP"

Set TIME5 to cause program interrupt #2 in 3.1 seconds

T5PHASE = 16074 (i.e. positive non-zero)

Set bit 3 of RCSFLAGS = 1

T5LOC = "RCSATT"

Set bits 15-14 (DAPBIT1, DAPBIT2) of FLAGWRD6 = 01_2

Perform "TVCZAP"

MMNUMBER = 0

Release interrupts

Delay 0.05 seconds (by putting job to sleep via "DELAYJOB")

Channel 5 = 0

Channel 6 = 0

Proceed to "ISITPOO"

INITSUB (Entered from "GOTOPOOH" and "ROO")

Inhibit interrupts

Set bit 1(P29FLAG) of FLAGWRD0 = 0

Set bit 6(IDLEFAIL) of FLAGWRD1 = 0

Set bits 12(P21FLAG), 11(STEERSW), and 9(IMPULSW) of FLAGWRD2 = 0

Set bits 14(GLOKFAIL) and 9(POOFLAG) of FLAGWRD3 = 0

Set bits 12(P23CALIB), 10(NEWTFLAG), and 7(ENGONFLG) of FLAGWRD5 = 0

Set bit 13(STRULLSW) of FLAGWRD6 = 0

Set bits 13(IGNFLAG), 12(ASTNFLAG), and 11(TIMRFLAG) of FLAGWRD7 = 0

Set bits 15(SWTOVER) and 11(V94FLAG) of FLAGWRD9 = 0

Proceed to "INITSUBA"

INITSUBA (Entered from "INITSUB", "ROO", and "TRACKTRM")

Inhibit interrupts

If bit 5(TRACKFLG) of FLAGWRD1 = 0:

Perform "STOPRATE"

(If bit 5(TRACKFLG) of FLAGWRD1 = 0):

If bit 12(MAXDBFLG) of FLAGWRD9 = 1:

Perform "SETMAXDB"

If bit 12(MAXDBFLG) of FLAGWRD9 = 0:

Perform "SETMINDB"

Release interrupts

OPTIND = -1

Return (to caller of "INITSUB" or "INITSUBA")

<u>ISITPOO</u>

If MMNUMBER $\neq$ 0:

If bit 1(NODOV37) of FLAGWRD2 = 1:

Perform "ALARM" (pattern 1520_8)

Perform "RELDSP"

Proceed to "PINBRNCH"

$TS_1 = K_{nov37m}$ (Tag here "CHECKTAB")

If (bits 7-1 of $K_{prml_{TS_1}}$) $\neq$ MMNUMBER:

If (bits 7-1 of $K_{prml_{TS_1}}$) $>$ MMNUMBER:

Set bit 7(Operator error) of channel 11 = 1

Perform "RELDSP"

Proceed to "PINBRNCH"

If TS_1 = 0:

Set bit 7(Operator error) of channel 11 = 1

Perform "RELDSP"

Proceed to "PINBRNCH"

$TS_1 = TS_1 - 1$

Proceed to 7th line of "ISITPOO" (check next table entry)

MINDEX = TS_1

Set bit 15(V50N18FL) of FLAGWRD3 = 1

If bit 6(V37FLAG) of FLAGWRD7 = 1: (Average-G running)

Set bit 1(AVEGFLAG) of FLAGWRD1 = 0

End of job (goes to "CANV37" from "AVGEND")

Proceed to "CANV37"

CANV37

CADRFLSH+2 = "ROO" + 3

Set restart group 4 to phase 1 (4.1, causing "INITDSP" to enter "ROO" if a restart)

Proceed to "ROO"

ROO

Perform "INTSTALL"

Set bits 10 (Caution Reset) and 9(Test connector Outbit) of channel 11 = 0

Set bits 14(S4B Cutoff), 13(S4B Injection Sequence start), 11(Disengage optics DAC), 10(Zero optics), 8(TVC enable), 3(Enable star tracker, not used), and 2(Enable Optics CDU Error Counters) of channel 12 = 0

Set bits 9(not assigned) and 8(not assigned) of channel 13 = 0

Perform "INITSUB"

Perform "CLEARMRK"

STARIND = 0

Set bit 14(STIKFLAG) of FLAGWRD1 = 0 (V50N18FL = 1 from "ISITPOO" unless POO selected)

Set bit 3(Uplink Activity) of channel 11 = 0

Set bit 14(R21MARK) of FLAGWRD2 = 0

If MMNUMBER = 0:

Perform "RELDSP" (tag here "POOH")

PHSPRDT2 = 05000_8 (i.e. priority 05_8)

Set bit 1(NODOV37) of FLAGWRD2 = 0

Set restart group 2 to phase 5 (i.e. 2.5, causing "STATINT1" to be established with priority 05_8 by restart logic)

(If MMNUMBER = 0):

Set bits 8(IMUSE) and 7(RNDVZFLG) of FLAGWRDO = 0

Set bit 9(UTFLAG) of FLAGWRD8 = 0

DNLSTCOD = 0

If MMNUMBER > 0:

If bit 7(RNDVZFLG) of FLAGWRDO = 0: (Tag here "NOUVEAU")

If bit 9(UTFLAG) of FLAGWRD8 = 0:

Set bit 8(IMUSE) of FLAGWRDO = 0

DNLSTCOD = (bits 15-13 of $K_{prml_{MINDEX}}$), cycled left 3 places (i.e. to bits 3-1)

DSPFLG+2 = 76657_8 (includes setting bit 4 and 6, either one of which causes "NORMRET" to bypass establishing "PLAYJUM1")

Set bits 10(LMTRG), 7(UPDATFLG), and 5(TRACKFLG) of FLAGWRD1 = 0

Set bit 2(R67FLAG) of FLAGWRD8 = 0

Make restart groups 3, 5, and 6 inactive

If MMNUMBER = 0:

Make restart groups 1,3,4,5, and 6 inactive (leaving only group 2 set, to 2.5, for "STATINT1")

Perform "INITSUBA" (TRACKFLG now zero, so "STOPRATE" etc.)

MODREG = MMNUMBER

Proceed to "GOPROG2" (restart group 2 will cause "STATINT1" to be established)

If MMNUMBER = 20: (Tag here "RENDVOO")

If MODREG = 20:

Make restart group 1 inactive (Tag here "KILL20")

Perform "INITSUBA"

Make restart group 2 inactive

Proceed to "REV37"

(If MMNUMBER = 20):

If bit 9(UTFLAG) of FLAGWRD8 = 0:

If bit 7(RNDVZFLG) of FLAGWRD0 = 1:

Set bits 7(UPDATFLG) and 5(TRACKFLG) of FLAGWRD1 = 1 (Tag here "STATQUO")

Make restart group 4 inactive (leaves 1 and 2 set by e.g. "AVGEND")

MODREG = MMNUMBER

Proceed to "GOPROG2" ("RELDSP" effect done in "STARTSB2" cell resets)

If bit 9(UTFLAG) of FLAGWRD8 = 1:

Set restart group 2 to phase 5 (i.e. 2.5, to cause "STATINT1" to be established with priority 05_8 by restart logic) (Tag here "STATQUO1")

Set bit 5(TRACKFLG) of FLAGWRD1 = 1

Make restart group 4 inactive

MODREG = MMNUMBER

Proceed to "GOPROG2"

If bit 9(UTFLAG) of FLAGWRD8 = 1: (Tag here "POOFIZZ")

Make restart group 2 inactive

Proceed to "REV37"

If bit 7(RNDVZFLG) of FLAGWRD0 = 1:

Proceed to "REV37"

Make restart group 1 inactive

Perform "INITSUBA"

Make restart group 2 inactive

Proceed to "REV37"

<u>REV37</u>

CADRFLSH+2 = "V37XEQ" + 3 (Group 4.1 set from "CANV37")

Proceed to "GOPROG2" (restart logic will cause "V37XEQ" to be entered via "INITDSP")

V37XEQ

Inhibit interrupts

PHSPRDT4 = 13000_8 (i.e. priority 13_8)

MMTEMP = $K_{prml_{MINDEX}}$

BASETEMP = $K_{fcadm_{MINDEX}}$

TS_2 = (bits 15-11 of BASETEMP) + (bits 10-8 of MMTEMP, shifted right 7 places)

TS_3 = (bits 10-1 of BASETEMP) + 2000_8

Establish a VAC area job, with starting address given in 2CADR format by (TS_3, TS_2), and with priority of 13_8

TS = (bits 7-1 of MMTEMP) and perform "NEWMODEX"

Perform "RELDSP"

End of job

P06

Set bit 1(NODOV37) of FLAGWRD2 = 1

Inhibit interrupts

TIME2SAV = T_{now}

Perform "SCALPREP": if return to calling address +1, proceed to second line of "P06"
otherwise, proceed

SCALSAVE = TS_{dp}

Set bits 5(TRACKFLG) of FLAGWRD1, 15(DRIFTFLG) of FLAGWRD2, and 13(REFSMFLG) of FLAGWRD3 = 0

Set bit 8(IMUSE) of FLAGWRD0 = 0

Set bit 7(RNDVZFLG) of FLAGWRD0 = 0

Set bit 9(UTFLAG) of FLAGWRD8 = 0

Set bit 11(Enable Standby) of channel 13 = 1

Set restart group 4 to cause it to establish "POSTAND" with priority 20_8 after a restart (erasable memory restart)

$TS = 00062_8$

Proceed to "GOPERF1" if terminate, proceed to previous line
if proceed, proceed to previous line
otherwise, proceed to previous line

<u>POSTAND</u> Entered after a restart (e.g. power-up) due to "P06" settings for restart group 4

Set bit 11(Enable Standby) of channel 13 = 0

Inhibit interrupts

$T_{now} = 0$

Perform "SCALPREP": if return to calling address +1, proceed to second line of "POSTAND"
otherwise, proceed

TS = TS - SCALSAVE, rescaled to B28 centi-seconds, with sign agreement forced

If $TS \leq -0$:

$TS = TS + 2^{23}$ centi-seconds (correct for channel 3 overflow)

TS = TS + TIME2SAV, with sign agreement forced

$T_{now} = T_{now} + TS$

Set bit 1(NODOV37) of FLAGWRD2 = 0

Proceed to "GOTOPOOH" (where restart group 4 is set)

<u>SCALPREP</u>

TS_{dp} = (Channel 3, Channel 4) Channel 4 read twice, then a 3rd time if 2 readings disagree. The channel 3 reading not done until channel 4 $\neq$ +MAX.

Release interrupts

$TS_{dp} = TS_{dp} + K_{rndbt}$

If bits 5-1 of TS+1 = 0, return to calling address +1 (Channel 4 was read in first interval after T_{now} was incremented)

Set bits 5-1 of TS+1 = 0

Return to calling address +2

<u>DELAYJOB</u>

i = 3

If DELAYLOC+i $\neq$ 0:

    If i = 0:

        Proceed to "BAILOUT" (pattern 31104_8)

    i = i - 1

    Proceed to second line of "DELAYJOB"

Call "WAKER" in (delay time) centi-seconds, with EBANK = i

DELAYLOC+i = Return address (to routine calling "DELAYJOB")

Put present job to sleep (starting address id = DELAYLOC+i)

<u>WAKER</u>

i = EBANK (mechanized by using BBANK with FBANK of "WAKER" = 00)

Set DELAYLOC+i = 0 and TS = DELAYLOC+i

Awaken job with starting address id = TS

End of task

<u>ALARM</u>

Inhibit interrupts

ALMCADR = "Calling address +1" (S-register portion)

Proceed to "ALARM2"

<u>ALARM2</u>

LREG = Alarm pattern (quantity in calling address +1)

ALMCADR+1 = BBANK + SUPERBNK (or'ed into bits 7-5)

If FAILREG+0 = 0:

    FAILREG+0 = LREG

    Skip next 2 lines

If FAILREG+1 = 0:

    FAILREG+1 = LREG

FAILREG+2 = LREG

Set bit 9(Program alarm) of DSPTAB+11 = 1, and flag for output
at next opportunity

Release interrupts

Return (to routine calling "ALARM" or "ALARM2")

PRIOLARM (Entered from "COM52" for alarm 0404_8)

Inhibit interrupts

LREG = TS (alarm pattern)

ALMCADR = "Calling address +1" (S-register portion)

ALMCADR+1 = BBANK of caller + SUPERBNK (or'ed into bits 7-5)

Perform "ALARM2" (starting at 3rd line)

TS = 0509_{vn}

Proceed to "PRIODSPR" (will return to routine calling "PRIOLARM")

VARALARM

Inhibit interrupts

LREG = TS

ALMCADR = "Calling address +1" (S-register portion)

Perform "ALARM2" (starting at 2nd line)

Return (address in ALMCADR)

BAILOUT

Inhibit interrupts

ALMCADR = "Calling address +1" (S-register portion)

Perform "VAC5STOR"

LREG = Alarm pattern (quantity in calling address +1)

Perform "ALARM2" (starting at second line)

Inhibit interrupts

If in interrupt mode, return to job status (set appropriate address into cell 0017_8 and then do a Resume)

Proceed to "ENEMA"

POODOO

Inhibit interrupts

ALMCADR = "Calling address +1" (S-register portion)

Perform "VAC5STOR"

LREG = Alarm pattern (quantity in calling address +1)

Perform "ALARM2" (starting at second line)

If bit 6(V37FLAG) of FLAGWRD7 = 0: (i.e. Average-G not running)

If (bits 13-1 of EXTVBACT) = 0: (error if bit 15 = 1)

Set bit 5(STATEFLG) of FLAGWRD3 = 0

Set bit 7(AUTOSEQ) of FLGWRD10 = 0

Set bit 13(INTGRAB) of FLGWRD10 = 0

Set bit 9(UTFLAG) of FLAGWRD8 = 0

Set bit 7(RNDVZFLG) of FLAGWRD0 = 0

Set bit 5(TRACKFLG) of FLAGWRD1 = 0

Set bit 1(NODOV37) of FLAGWRD2 = 0

Make all restart groups inactive (by setting PHASEi = +0 and -PHASEi = -0 for i = 1-6)

Inhibit interrupts

If in interrupt mode, return to job status (set appropriate address into cell 0017_8 and then do a Resume)

Proceed to "ENEMA"

SVCT3 (This task is used as part of the waitlist control, and is entered every 81.93 seconds for that purpose)

If bit 15(DRIFTFLG) of FLAGWRD2 = 0:

End of task

If IMUCADR = +0:

Establish "NBDONLY" (priority 35_8)

End of task

Delay 5 seconds

Proceed to "SVCT3"

DUMMYJOB — This routine is entered from the executive system if no active jobs are found.

NEWJOB = -0

Release interrupts

Set bit 2(Computer Activity) of channel 11 = 0

Proceed to second line of "CHECKNJ"

CHECKNJ

SELFRET = Return address

If NEWJOB = -0:

Return (to address specified by SELFRET, BBCON of "SELFCHK")

Set bit 2(Computer Activity) of channel 11 = 1

Perform functions necessary to start new job's computations

Quantities in Computations

See also list of major variables and list of routines

1dPIPADT: See IMU Computations.

AIG: See Digital Autopilot Entry Routines.

AIGdPIP: Value of AIG sampled in "READACCS", i.e. the value of CDU_y sampled about 0.05 seconds before accelerometers read, scale factor B-1, units revolutions, single precision. Program notation is "AIG/PIP".

ALFAd180: See Entry Computations (quantity is updated in "ATTRATES").

ALFAdPIP: Value of ALFAd180 sampled in "READACCS", scale factor B-1, units revolutions, single precision. It is computed based upon the same angle sample as for AIGdPIP. Program notation "ALFA/PIP".

ALMCADR, ALMCADR+1: Pair of erasable memory cells used to store the information on the calling address to the "ALARM" type routines (including "BAILOUT" and "POODOO"). Information is in 2CADR format (see 3420.5-27), with ALMCADR giving S-register information and ALMCADR+1 giving BBCON (i.e. BBANK and SUPERBNK) information. Cells can be displayed by N08 to permit identification of the specific area of the program responsible for the generation of the fault condition.

AMG: See Digital Autopilot Entry Routines.

AMGdPIP: Value of AMG sampled in "READACCS", i.e. the value of CDU_z sampled about 0.05 seconds before accelerometers read, scale factor B-1, units revolutions, single precision.

AOG: See Digital Autopilot Entry Routines.

AOGdPIP: Value of AOG sampled in "READACCS", i.e. the value of CDU_x sampled about 0.05 seconds before accelerometers read, scale factor B-1, units revolutions, single precision.

AVEGEXIT: Cell containing in 2CADR format the starting address of the computations to be performed after "AVERAGEG" is done (program notation also AVGEXIT). It is set to "AVGEND" by "READACCS"; to "CALCN83" by "TIGON"; to "CALCN85" by "P40RCS" and "P40SXTY"; to "CM/POSE" by "P62"; to "SERVEXIT" by "P67.1", "POST41", "S61.1C", and "TIGNOW"; to "SIVBCOMP" by "P15JOB"; to "S40.8" by "P40SXTY"; and to "VHHDOT" by "P11".

BASETEMP: Cell used in "V37XEQ" to store temporarily the appropriate value of K_{fcadm} for use in generating starting address information for the job to be initiated by V37 processing.

BBANK: A computer hardware cell (address 0006_8) containing in bits 15-11 the fixed memory bank (FBANK) currently being used and in bits 3-1 the erasable memory bank number. Bits 7-5 of the constants in the proper format for BBANK loading can contain SUPERBNK information, and constants in this format are referred to as "BBCON" format (see 3420.5-27). The fact that the program retains BBANK information is used in "DELAYJOB" to retain the indexing data for the DELAYLOC cell employed (since in that routine the erasable memory bank is not significant and the fixed memory bank number is 00).

BETAd180: See Entry Computations (quantity is updated in "ATTRATES").

BETAdPIP: Value of BETAd180 sampled in "READACCS", scale factor B-1, units revolutions, single precision. Cf. ALFAdPIP.

$\underline{C}_{unitw}$: Erasable memory vector constant, scale factor B0, program notation "UNITW". It gives the polar axis in the reference coordinate system. The x component (also called "-AYO" in the program) gives the "true to mean pole rotation about the -Y axis;" the y component (also called "AXO" in the program) gives the "true to mean pole rotation about the +X axis."

CADRFLSH+2: See Display Interface Routines.

CADRSTOR: See Data Input/Output.

CLPASS: See Data Input/Output.

CMdGYMDT: See Digital Autopilot Entry Routines.

DELAYLOC+i (i = 0 - 3): Set of four single precision cells used to contain the starting address identifications for jobs put to sleep using "DELAYJOB", in FCADR format. If an attempt is made to put more than 4 jobs to sleep using the routine, a software restart (pattern 31104_8) is produced.

DELVREF: Value of accelerometer output (scaled and compensated) converted to reference coordinates, scale factor B7, units meters/centi-second.

DNLSTCOD: See Telemetry.

DNTMGOTO: See Telemetry.

DSPCNT: Single precision counter used to cycle through the $DSPTAB_i$ (i = 10-0) to be changed in "DSPOUTSB", scale factor B14. The $DSPTAB_i$ word loaded at one entrance to "DSPOUTSB" is the first one checked at the next entrance, but that word will not be checked again until all the other words have been examined and, if necessary, loaded into OUTO.

DSPCOUNT: See Data Input/Output.

DSPFLG+2: See Display Interface Routines.

DSPLIST, DSPLOCK: See Data Input/Output.

DSRUPTSW: Single precision cell that is used for control of the computations that are performed when "T4RUPT" is entered. If it is not negative non-zero when "T4RUPT" is entered, it contains a number between 3 and 0, scale factor B14, used to control the branching at the end of "PROCEEDE" so as to complete a cycle (3 through 0 and back to 3 again) in 0.48 seconds. If the cell is negative non-zero, bits 2-1 contain the same "T4RUPT" branching control ("PROCEEDE" is entered and these bits employed every 0.12 seconds), while bits 12-9 contain, scale factor B6, the complement of the number of 20-millisecond intervals still be to loaded into TIME4 to achieve synchronism with the basic T4RUPT period of 0.12 seconds. Bit 14 is used to achieve an alternate selection of clearing OUTO or loading new information into OUTO. If OUTO was loaded with a non-zero value when the basic 0.12-second T4RUPT cycle was performed, then DSRUPTSW is set (in "HANG20") to DSRUPTSW - 22400_8, where the first "2" causes bit 14 to be set 0 (note that computation same as adding 55377_8), and the "24" is 5 with scale factor B6, signifying that five more 20-ms loadings of TIME4, plus the current one, are needed to satisfy the 0.12 second synchronism. If "HANG20" is entered, TIME4 is set to generate another interrupt in 0.02 second. When it is received, bit 14 of DSRUPTSW will be 0, so OUTO will be cleared, bit 14 set to 1, and TIME4 reset to 20 ms. In addition, DSRUPTSW is incremented by K_{1b6} (making the upper bits equivalent to -4B6, since bit 14 = 1, for a negative DSRUPTSW, represents a magnitude of 0). The next time the interrupt is generated, bit 14 will be one in "T4RUPT", so a new OUTO can be generated if required (if not, the number of remaining 20-ms intervals is determined and TIME4 and DSRUPTSW loaded immediately). In order to load all 11 (DSPTAB+0 - DSPTAB+10) DSKY display registers, a total time of 11 x 0.04 = 0.44 seconds is required (measured from the first loading of the first register to the end of the 20 ms wait after reset of OUTO for the last register). As part of a verb 36 fresh start, DSRUPTSW = - K_{12b5}, to force clearing of all DSPTAB-driven relays (bit 14 = 1 due to this setting, of course) before starting the normal T4RUPT 0.12-second cycle.

DVTOTAL: Sum of magnitudes of scaled and compensated $\underline{DELV}$ each two-second entrance to "SERVICER", scale factor B7, units meters/centi-second. Quantity is initialized to 0 in "PREREAD1", and can be displayed in R3 of N40 (XXXX.X fps) or N80 (XXXXX. fps). It is the "line integral" of accumulated velocity, not the magnitude of the velocity vector increment: the magnitude is taken each two-second Average-G cycle of $\underline{DELV}$ and this scalar is added to DVTOTAL.

EBANK: See Data Input/Output.

ERCOUNT: See Testing Routines.

ERESTORE: See Testing Routines.

EXTVBACT: See Verb Definitions.

FAILREG+i (i = 0 - 2): Set of three single precision cells used to retain alarm pattern code information. They are all reset to zero by a verb 36 fresh start (or V25N9E E E E), and they can be displayed by the use of noun 09. Use of the error reset DSKY key causes FAILREG+0 and FAILREG+1 to be reset zero (FAILREG+2 is left alone). FAILREG+0 contains the first alarm pattern received after the reset; FAILREG+1 contains the second; and FAILREG+2 always has the most recent (and hence would be the same as FAILREG+0 if one alarm has occurred, or FAILREG+1 if only two alarms have occurred). If FAILREG+0 = 0, then no alarms have occurred since the last error reset, and the alarm to occur most recently is in FAILREG+2.

GD$\underline{T}$: Value of gravity times computing interval, scale factor B8, units meters/centi-second. Program notation is "GDT/2", in recognition of the fact that it is "$\frac{1}{2}$ GD$\underline{T}$" for a standard velocity scaling of B7 meters/centi-second. When "CALCRVG" is first entered, the quantity in GD$\underline{T}$ is the value computed the previous computation cycle.

GDT$\underline{1}$: Value of gravity times computing interval computed in "CALCGRAV", same scaling and units as GD$\underline{T}$ (program notation "GDT1/2").

GOB$\underline{L}$: Value of "normalized" oblateness component of gravity computed for earth-centered computations only. Must be multiplied by $-\mu/r^2$ to convert to units of acceleration. It is used in "S40.9" to correct output of Lambert routine (for earth-centered computations). Program notation "GOBL/2". Loaded with GOBL$\underline{1}$ whether earth-centered or moon-centered, but not used in "S40.9" for moon-centered. Scale factor B1, dimensionless.

GOBL$\underline{1}$: Value of GOB$\underline{L}$ computed in "CALCGRAV", same scaling as GOB$\underline{L}$ (program notation "GOBL1/2").

IMODES30, IMODES33: See IMU Computations.

IMUCADR: See IMU Computations.

INLINK: See Data Input/Output.

K_{1b6}: Single precision constant, program notation "BIT9", scale factor B6, value 00400_8, corresponding to 1×2^{-6}, used to increment DSRUPTSW for 20-ms OUTO update rate (cf. DSRUPTSW).

K_{2j}: Constant, program notation "2J", scale factor B-1, value $3.2469201\text{E-}3 \times 2^1$. Value corresponds to $2 \times 1.62346005\text{E-}3 \times 2^1$, where first term is an equation factor of 2, second is the nominal value of the computer-employed first gravitational harmonic coefficient (note that the "standard" value, such as that employed in K_{j2} of Orbital Integration, is 1.6234$\underline{5}$E-3), and third is scale factor. Octal value corresponds to a harmonic coefficient value of about 1.6234554E-3.

K_{2s22}: Constant, program notation "2SEC(22)", scale factor B22, units centi-seconds. Value is 200 x 2^{-22}, corresponding to a value of 2 seconds.

K_{12b5}: Single precision constant, program notation "PRIO12", scale factor B6, value 12000_8, corresponding to 24 x 2^{-6}. It is used to initialize DSRUPTSW to ensure clearing of all DSPTAB registers for a V36 fresh start (see DSRUPTSW): it forces 12 sets of OUTO settings (DSPTAB+11 and then remaining DSPTAB's) and subsequent OUTO = 0 settings before starting the normal T4RUPT 0.12-second cycle, hence the notation.

K_{20j}: Constant, program notation "20J", scale factor B-1, value 3.2469201E-2 x 2^1. Value corresponds to 20 x 1.62346005E-3 x 2^1, where first term is an equation factor of 20 (to give a net effect of "1 - 5 SINØ²" for the factor it multiplies), second is the nominal value of the computer-employed first gravitational harmonic coefficient (cf. K_{2j}), and third is scale factor. Octal value corresponds to a harmonic coefficient value of about 1.62346009E-3.

K_{fcadm_i}: Table of fixed-memory starting addresses used in "V37XEQ" to set BASETEMP, arranged in FCADR format (bits 15-11 give the fixed-memory bank and bits 10-1, when added to 2000_8, give S-register contents). Program notation for first cell in table (i = 0) is "FCADRMM1".

K_{mmudt_0}: Constant, program notation "-MUDT(E)", scale factor B44, units meters³/centi-second. Value is -7.9720645E12 x 2^{-44}, corresponding to -1 x 3.98603225E10 x 200 x 2^{-44}, where first term is an equation factor to give proper sign of result, second is value of μ (for time in centi-seconds) for the earth, third is computing interval of two seconds expressed in centi-seconds, and fourth is scale factor.

K_{mmudt_2}: Constant, program notation "-MUDT(M)", scale factor B44, units meters³/centi-second. Value is -9.805556E10 x 2^{-44}, corresponding to -1 x 4.902778E8 x 200 x 2^{-44}, where first term is an equation factor to give proper sign of result, second is value of μ (for time in centi-seconds) for the moon, third is computing interval of two seconds expressed in centi-seconds, and fourth is scale factor.

K_{mxdlv}: Single precision constant, program notation "-MAXDELV", scale factor B14, units accelerometer counts. Stored value is -6398 x 2^{14}; used in program in such a way that an effective value of 6399 is employed, giving an alarm (at the start of "SERVICER") if an accelerometer output of 6400 pulses for 2 seconds (i.e. 3200 pps) is received.

K_{nov37m}: Single precision constant, program notation "NOV37MM", scale factor B14, giving the number of table entries minus 1 in K_{prml_i} (and associated V37 tables) for new programs that can be started by the V37 logic (including the minimum key rendezvous driver programs, but excluding P00). Value is 00046_8, corresponding to decimal 38 (meaning 39 programs).

K_{ppl}: Constant, program notation "KPIP1", scale factor B-7, units meters/centi-second per accelerometer count. Value is 0.07488, corresponding to $5.85 \times 10^{-2} \times 10^{-2} \times 2^7$, where first term is nominal accelerometer scale factor in units of centimeters/second per count, second converts to meters, third converts to centi-seconds, and fourth is scale factor.

K_{prml_i}: Table of program information used in association with V37 program changes. Table is arranged in order of decreasing program numbers (a higher i is a lower number), with first cell having program notation "PREMM1". Bits 7-1 of the cell give the program number (since program numbers are in decimal, seven bits are necessary to represent numbers above 63); bits 10-8 are the erasable memory bank number (required when the job is established); and bits 15-13 are the job's required DNLSTCOD setting for telemetry. See information below.

K_{resq}: Constant, program notation "RESQ", scale factor B59, units meters^2. Nominal value is $40.6809913E12 \times 2^{-59}$, where the decimal portion corresponds to the square of 6 378 165.2 meters. Octal value is 00001_8 05000_8, however, which corresponds to the square of about 6 378 238.8 meters.

K_{rndbt}: Single precision constant, program notation "BIT5",scale factor (as used) B23, units centi-seconds. Value is 00020_8, corresponding to (as used, i.e. added to the least significant half of channel 4 time information) 0.5 centi-seconds. Used to account for the fact that channel 4 bits 5-1 "read 20_8 for the first interval after a TIME1 increment", thus causing bit 6 of channel 4 to be incremented 5 ms out of phase with TIME1 increments.

K_{rtu_i} (i = 0 - 11): Table of single precision constants, program notation "RELTAB", containing in bits 15-12 the required values to be loaded into bits 15-12 of OUT0 for $DSPTAB_i$. Value in these bits is (i + 1). Bit 11 is zero. The least significant 5 bits of these same constants are used for K_{rtb_I} (see Data Input/Output).

LGYRO: See IMU Computations.

LREG: Computer "L" register (address 0001_8), used to retain alarm pattern information in alarm-generation package.

MARKINDX: See Optics Computations.

MINDEX: Single precision cell, scale factor B14, used to select the appropriate table entries for a V37-selected program change (loaded based on equality of MMNUMBER and bits 7-1 of K_{prml_i} with the value of i).

MMNUMBER: Single precision cell, scale factor B14, used to retain the desired new program number for a V37 program change (it is the number entered in "MMCHANG" or is 0 if "V37" entered from "VERB96"). If the TVC DAP or SPS engine-on are indicated, it is set 0.

MMTEMP: Single precision cell loaded in "V37XEQ" with the value of the appropriate K_{prml} for use in generating appropriate information in connection with the job to be established by V37 processing.

MONSAVE, MONSAVE1: See Data Input/Output.

MRKBUF1: See Optics Computations.

NEWJOB: Cell set positive non-zero to indicate that a job of higher priority than the one presently being performed (the "DUMMYJOB" loop has minimum priority) has been established and is awaiting execution. Cell 0067_8 is used for the word, and must be addressed periodically in order to avoid generating a hardware restart (see hardware documentation for details).

NOUNREG: See Data Input/Output.

NOUT: Single precision cell, scale factor B14, used to provide a count of the number of DSPTAB display outputs (DSPTAB+0 - DSPTAB+10) that are remaining to be changed: if it is non-zero but none are to be changed, it is set 0.

NVWORD+2: See Display Interface Routines.

OPTIND: See Optics Computations.

OPTMODES: See Optics Computations.

OUTO: Computer output channel 10, used to transmit relay driving information to the display system. Bits 15-12 specify the group of relays to be driven, while bits 11-1 specify their new state.

PHASEi, -PHASEi (i = 1 - 6): Set of cells used to contain program octal "phase" information, used to control program restarts (each "i" is called a "group"). To check the validity of erasable memory information (and the phase information itself), a check is made that the information in PHASEi, when complemented, is the same as the information in -PHASEi (for all i). A setting of +0 for PHASEi means that the group restart controlled by that cell is "inactive" (see Section VIIC of 3420.5-27 for details of restart logic computations not covered in this writeup). In the interests of avoiding excessive detail (since the "normal" mode of operation of the computer system does not require restarts), most of the settings for restart groups are not shown in the writeups, although settings in some cases form an integral part of the program control logic, and hence are shown in such cases.

As a general guide to allocation of restart groups, the following areas where each group is used are supplied as a partial list:

#1: P20 (setting to 1.11 causes "PIKUP20" to be entered). A setting to 1.7 is done in "CHKLINUS" for R61-specified R60 maneuver. A setting to 1.5 is used to protect "S40.9" (set by "SETUP.9"), which of course only occurs with Average-G on, when P20 doesn't run. If "R00" concludes that P20 should not run, it resets group 1 to inactive, thus over-riding the "AVGEND" setting.

#2: Orbital integration (including the periodic state vector update started by "STATINT1" for a 2.5 setting). A setting of 2.7 causes "R22" to be established, and 2.13 causes "REDOR22" to be established (the latter setting is done by "AVGEND", and written over by "R00" if necessary). Group 2 is also used for protection within the R22 computations, which are done at sufficiently high priority that they would not be expected to be interrupted by mission programs except while R22 is performing "WAITONE". Restart priority lower than that at which job runs to permit mission programs to set e.g. TRACKFLG. A 2.11 setting is done by "VERB94" for P23 computations (to permit "V94ENTER" to be established while getting rid of e.g. R52 if it happens to be running in P23 then).

#3: "S40.13"; "P11"; "ENGINOFF".

#4: All V37-selectable programs except P00 (and also other mission program sequences running outside of Average-G cycle, such as "P65.1"); "GCOMPVER"; "AZMTHCG1". Also V37 processing itself.

#5: "READACCS" (and subsequent "SERVICER" computations); Prelaunch alignment.

#6: "CLOKTASK"; Entry autopilot; P27; "S40.6" (via "PRE40.6").

PHSPRDT2: Single precision erasable memory cell used to contain priority or time information for erasable memory restarts using group 2 (see Section VIIC of 3420.5-27). It is set to indicate priority 05_8 for P00 selection in "R00" in order to achieve proper priority information for group 2 use in the periodic P00 orbital integration.

PHSPRDT4: Single precision erasable memory cell used to contain priority or time information for erasable memory restarts using group 4 (see Section VIIC of 3420.5-27). It is set to the priority of the mission program that is established by "V37XEQ", for subsequent use by erasable memory restarts.

PIPA: See IMU Computations.

PIPCTR: Single precision cell, scale factor B14, used to control the performance of "QUIKREAD". It is set to 2 in "READACCS" to allow "QUIKREAD" to be performed 1.0 and 1.5 seconds after "READACCS" ("READACCS" itself causes "QUIKREAD" to be done 0.5 seconds after "READACCS").

PRIORITY$_i$ (i = 0 - 6): Set of cells containing in bits 14-10 the priority information on computer jobs associated with the job register set (see Section VIIB of 3420.5-27). If the cell contains -0, this indicates that the corresponding job register set is available.

QREG: Computer hardware address 0002_8, loaded with return address information after a "TC" machine language transfer instruction. See 3420.5-27 for details (program notation "Q"): the S-register portion is loaded into QREG.

$\underline{R}_{nl}$: Temporary storage for updated value of $\underline{R}$ in "CALCRVG", used for restart protection purposes and to ensure a homogeneous state vector on the downlink (same units and scaling as $\underline{R}$). It is used in "NORMLIZE" to initialize $\underline{R}$ for Average-G: in such cases, $\underline{R}_{nl}$ loaded either by "P11" or "MIDTOAV2".

RCSFLAGS: See Digital Autopilot Interface Routines.

REDOCTR: Single precision cell, scale factor B14, used to count the number of hardware restarts (entrances to "GOPROG") that have been performed. It is initialized to 0 as part of a verb 36 fresh start.

REQRET: See Data Input/Output.

RESTREG: See Display Interface Routines.

ROLLd180: See Entry Computations: quantity updated in "ATTRATES".

ROLLdPIP: Value of ROLLd180 sampled in "READACCS", scale factor B-1, units revolutions, single precision. See ALFAdPIP.

RSBBQ: Value of address where hardware restart (causing entrance to "GOPROG") was generated. Most significant half contains BBCON (see BBANK above) information and least significant half has Q-register information.

RTX2: See Orbital Integration. It is left at 0 for earth and 2 for moon by "MIDTOAV2" (used to initialize $\underline{R}_{nl}$ etc. for Average-G), and is set 0 in "P11". Used e.g. in "CALCGRAV" to determine proper gravity formula to be used.

SCALSAVE: Value of (Channel 3, Channel 4) saved in "P06", scale factor B23, units centi-seconds. Used to restore the proper value of the computer clock (modulo 2^{23} centi-seconds or about 23.3 hours) after a period of low-power operation.

SELFRET: Cell containing return address information to routine calling "CHECKNJ" (a location in the computer self-check memory bank, hence the name). Must be preset as part of the initial conditions routine so that the "DUMMYJOB" routine will function properly.

SINØ: Value of sine of latitude computed in "CALCGRAV", scale factor B1.

SKEEPi: See Testing Routines.

SMODE: See Testing Routines.

SOLD, SRATE: See Optics Computations.

STARIND: See Inflight Alignment.

SUPERBNK: Value of computer channel 07, containing in bits 7-5 the specification of the "super bank" setting: value of 0-3 causes FBANK settings of 30_8-37_8 to read out correspondingly numbered fixed memory banks; a setting of 4 causes FBANK settings of 30_8-33_8 to read out fixed memory banks 40-43 (used e.g. for most DSKY processing). Also referred to as FEXT.

T_{pptml}: See IMU Computations.

T5LOC: See Digital Autopilot Interface Routines.

T5PHASE: See Digital Autopilot RCS Routines.

TEMPMM: See Minimum Key Rendezvous.

TIME2SAV: Value of T_{now} retained in "P06", scale factor B28, units centi-seconds (the corresponding value of channels 3 and 4 is in SCALSAVE), used to permit restoration of the proper value of the computer clock (T_{now}) after a period of low-power operation.

TIME3, TIME4, TIME5: Computer hardware erasable memory counter clock cells, whose overflow cause program interrupts #3, #4, and #2 respectively (all are scale factor B14 in units of centi-seconds, and are set to (16384 - t), with t in centi-seconds, to cause overflow in "t" centi-seconds). See 3420.5-27.

TOLD, TRATE: See Optics Computations.

TRKMKCNT: See Measurement Incorporation.

TTE, TTE2: See Display Computations.

UPSVFLAG: See Uplink Processing.

$\underline{V}_{nl}$: Temporary storage for updated value of $\underline{V}$ in "CALCRVG", used for restart protection purposes and to ensure a homogeneous state vector on the downlink (same units and scaling as $\underline{V}$). It is used in "NORMLIZE" to initialize $\underline{V}$ for Average-G: in such cases, $\underline{V}_{nl}$ is loaded either by "P11" or "MIDTOAV2".

VAC$\underline{I}$USE ($\underline{I}$ = 1-5): Control cell indicating, if zero, that the associated VAC area is assigned; otherwise, it contains its own address (hence VAC3USE +1 would be the address of the first available cell in the 43-cell VAC area). Quantities indicated as in "push-down list", or by their "relative address" identification, are stored in a VAC area. See 3420.5-27 for more details.

VERBREG: See Data Input/Output.

VHFCNT: See Measurement Incorporation.

WTOPTION: See Optics Computations.

XOLDBU$\underline{F}$: Value of previous XPIPBU$\underline{F}$, containing accelerometer samples with scale factor B14, units pulses. Cells are all single precision, with program notations XOLDBUF, YOLDBUF, and ZOLDBUF respectively. Use same cells as $ADOT_1$+1, $ADOT_{2sp}$, and $ADOT_2$+1 (for telemetry purposes): see Digital Autopilot RCS Routines.

XPIPBU$\underline{F}$: Set of three consecutive erasable memory cells, scale factor B14, units pulses (accelerometer counts, uncompensated). Cells are all single precision, with program notations XPIPBUF, YPIPBUF, and ZPIPBUF respectively. Use same cells as $ADOT_{0sp}$, $ADOT_0$+1, and $ADOT_{1sp}$ (for telemetry purposes): see Digital Autopilot RCS Routines. If Entry DAP is running(bit 12 of FLAGWRD6 = 1), loaded in "READACCS" with raw accelerometer sample, and updated 0.5, 1.0, and 1.5 seconds later with present accelerometer count (in "QUIKREAD"), with previous value of XPIPBU$\underline{F}$ being stored in XOLDBU$\underline{F}$.

Verb 37 Major Mode Tables

Mode	Address	EBANK	DNLSTCOD	Mode	Address	EBANK	DNLSTCOD
00	---	-	0	47	"P47CSM"	7	3
01	"GTSCPSS"	5	0	51	"P51"	5	0
06	"P06"	4	0	52	"PROG52"	5	0
15	"P15JOB"	7	3	53	"P51"*	5	0
20	"PROG20"	6	2	54	"PROG52"*	5	0
21	"PROG21"	4	2	61	"P61"	6	3
22	"PROG22"	7	4	62	"P62"	6	1
23	"P23"	7	2	72	"P72"	4	2
24	"PROG24"	5	4	73	"P73"	4	2
29	"P29"	7	2	74	"P74"	4	2
30	"P30"	7	2	75	"P75"	4	2
31	"P31"	4	2	76	"P76ER77"	7	2
32	"P32"	4	2	77	"P76ER77"	7	2
33	"P33"	4	2	79	"P79"	7	2
34	"P34"	4	2	81	"P81"	6	2
35	"P35"	4	2	82	"P82"	6	2
36	"P36"	4	2	83	"P83"	6	2
37	"P37"	7	2	84	"P84"	6	2
40	"P40CSM"	6	3	85	"P85"	6	2
41	"P41CSM"	6	3	86	"P86"	6	2

*Table entries for P53 and P54 indicate starting addresses of "P53" and "P54" respectively, but these tags are defined to be equal to "P51" and "PROG52", as shown.

P81-P86 are control drivers for minimum key rendezvous cycling: they are not intended for manual selection.

IMU Computations

RO2BOTH Entered from "P40S/F" (P40/P41), "P47CSM", "PROG20", "PROG22", "PROG24", "PROG52" (P52/P54), "S61.1" (P61/P62), and "V89CALL"

If bit 13(REFSMFLG) of FLAGWRD3 = 0:

If bit 9(IMUOPBIT) of IMODES30 = 0: (IMU turned on)

$TS = 0220_8$

If bit 9(IMUOPBIT) of IMODES30 = 1:

$TS = 0210_8$

Perform "VARALARM"

Proceed to "GOTOPOOH"

Set bit 8(IMUSE) of FLAGWRD0 = 1

Return

PIPASR

$T_{pptm1} = T_{now}$

Set least significant halves of $DEL\underline{V}$ = 0 (flag for telemetry that $DEL\underline{V}$ not compensated)

Set $DEL\underline{V} = PIP\underline{A}$ and $PIP\underline{A} = -0$ — $PIP\underline{A}$ reset so that no counts are lost. Only most significant halves of $DEL\underline{V}$ loaded. Special precautions taken to make routine restartable: if get a restart, 1dPIPADT set to K_{2secp}.

Return

1/PIPA

If GCOMPSW < 0, Return

Inhibit interrupts

Perform the following for i = z, y, x:

$$DELV_i = DELV_{i_{sp}} + C_{pipascf_i}\, DELV_{i_{sp}} - C_{pipabias_i}\, 1dPIPADT$$

Release interrupts

$$GCOMP_x = GCOMP_x - C_{ad,x}\, DELV_x + C_{sr,x}\, DELV_y - C_{nbd,x}\, 1dPIPADT$$

$$GCOMP_y = GCOMP_y - C_{ad,y}\, DELV_y + C_{sr,y}\, DELV_z - C_{nbd,y}\, 1dPIPADT$$

$$GCOMP_z = GCOMP_z - C_{ad,z}\, DELV_z - C_{sr,z}\, DELV_y + C_{nbd,z}\, 1dPIPADT$$

If magnitude of any $GCOMP_i$ $(i = x,y,z) \geq K_{cpck}$:

Establish "1/CHECK" (priority 21_8)

Return

NBDONLY Established by "SVCT3"

If GCOMPSW < 0, End of job

Inhibit interrupts

If bit 15(DRIFTFLG) of FLAGWRD2 = 0, End of job

Set TS = 1dPIPADT and 1dPIPADT = TIME1

Release interrupts

TS = 1dPIPADT - TS (i.e. new value minus old value)

Proceed to "NBD2"

LASTBIAS Established by "PREREAD"

Perform "PIPUSE"

If GCOMPSW < 0, End of job

TS = (least significant half of T_{pptm1}) - 1dPIPADT

1dPIPADT = K_{2secp}

Proceed to "NBD2"

NBD2

If TS = 0, End of job

If TS < 0:

TS = TS + 16384 centi-seconds (correct for TIME1 overflow)

$GCOMP_x = GCOMP_x - C_{nbd,x}$ TS (TS scaling here B19)

$GCOMP_y = GCOMP_y - C_{nbd,y}$ TS

$GCOMP_z = GCOMP_z + C_{nbd,z}$ TS

If magnitude of any $GCOMP_i$ $(i = x,y,z) \geq K_{cpck}$:

Proceed to "1/GYRO"

End of job

1/CHECK

If IMUCADR $\neq$ 0:

End of job

Proceed to "1/GYRO"

1/GYRO

Shift GCOMP right 7 places (bits corresponding to 2^{-8} pulses and below are lost), making scaling B21 pulses

TS = "GCOMP"

Perform "IMUPULSE"

Perform "IMUSTALL": if error return, proceed
otherwise, proceed

Shift least significant half of GCOMP left 7 places, making scaling of double precision number (for most significant halves 0) B14 pulses. Bits 14-8 of least significant half lost (expected to be 0 for normal operation)

End of job

IMUMON Entered from "PROCEEDE" every 0.48 seconds

TS_1 = (bits 15-11 and 9 of channel 30)

If TS_1 = (bits 15-11 and 9 of IMODES30):

Proceed to "TNONTEST"

TS = 0 (in RUPTREG2 cell)

Set those bits of TS = 1 that are different in IMODES30 and TS_1 (in the range of bits 15-11 and 9)

Set (bits 15-11 and 9 of IMODES30) = TS_1

If bit 15 of TS = 1: (IMU temperature)

TS = bits 14-1 of TS (Tag here "TLIM")

If bit 15(TLIMBIT) of IMODES30 = 1:

Set bit 4(Temperature Caution) of channel 11 = 1

If bit 15(TLIMBIT) of IMODES30 = 0:

If bit 1(LMPTSTBT) of IMODES33 = 0:

Set bit 4(Temperature Caution) of channel 11 = 0

If bit 14 of TS = 1: (IMU turn-on request)

If bit 2(DLAYFAIL) of IMODES30 = 0: (Tag here "ITURNON")

If bit 14(TONISSBT) of IMODES30 = 0:

Set bit 7(ITNON1BT) of IMODES30 = 1

If bit 14(TONISSBT) of IMODES30 = 1:

If bit 15(ISS Turn-on Delay Complete) of channel 12 = 0:

Set bit 2(DLAYFAIL) of IMODES30 = 1

Perform "ALARM" (pattern 0207_8)

If bit 13 of TS = 1: (IMU fail)

Perform "SETISSW"

If bit 12 of TS = 1: (IMU CDU fail)

Perform "SETISSW"

If bit 11 of TS = 1: (IMU cage command)

If bit 11(CAGEBIT) of IMODES30 = 1: (Tag here "IMUCAGE")

Set bit 4 (No Attitude) of DSPTAB+11 = 0, and flag for output at next opportunity

Set bit 5(Zero IMU CDU) of channel 12 = 1

CDU̲ = 0

Call "UNZ2" in 0.32 seconds

Proceed to "C33TEST"

Set bits 15-10 (IMU and CDU drive enable and gyro drive enable) of channel 14 = 0

Set bits 8(TVC Enable), 6-4 (Enable IMU CDU Error Counters, Zero IMU CDU, and Coarse Align IMU), and 2(Enable Optics Error Counters) of channel 12 = 0

Set bit 7(ENGONFLG) of FLAGWRD5 = 0

Set bit 13(SPS Engine On) of channel 11 = 0

Perform "CAGESUB" (starting at 3rd line)

(If bit 11 of TS = 1):

Set bits 5(TRACKFLG) of FLAGWRD1, 15(DRIFTFLG) of FLAGWRD2, and 13(REFSMFLG) of FLAGWRD3 = 0

CDUXCMD = -0

GYROCMD = -0

Set bits 9-6 (Gyro drive selection) of channel 14 = 0

If bit 9 of TS = 1: (IMU operate)

If bit 9(IMUOPBIT) of IMODES30 = 1: (Tag here "IMUOP")

Set bit 6(NOIMUDAP) of IMODES33 = 1

Set bits 5(TRACKFLG) of FLAGWRD1, 15(DRIFTFLG) of FLAGWRD2, and 13(REFSMFLG) of FLAGWRD3 = 0

Set bit 9(UTFLAG) of FLAGWRD8 = 0

If bit 8(IMUSE) of FLAGWRD0 = 1:

Perform "ALARM" (pattern 0214_8)

Set bits 8(IMUSE) and 7(RNDVZFLG) of FLAGWRD0 = 0

If bit 9(IMUOPBIT) of IMODES30 = 0:

If bit 2(DLAYFAIL) of IMODES30 = 0:

Set bit 7(ITNON1BT) of IMODES30 = 1

Proceed to "TNONTEST"

TNONTEST

If bit 7(ITNON1BT) of IMODES30 = 0:

Proceed to "C33TEST"

If bit 8(ITNON2BT) of IMODES30 = 0:

Set bit 8(ITNON2BT) of IMODES30 = 1

Proceed to "C33TEST"

Set bits 8-7(ITNON2BT, ITNON1BT) of IMODES30 = 0 (Tag here "PROCTNON")

If bit 14(TONISSBT) of IMODES30 = 1:

If bit 4(Coarse Align) of channel 12 = 1: (Tag here "OPONLY")

Proceed to "C33TEST"

If bit 8(IMUSE) of FLAGWRDO = 1:

Proceed to "C33TEST"

Set bits 6-3 (IMUNITBT, NOACCALM, IMUFINHT, ICDUINHT) and bit 1(ACCFINHT) of IMODES30 = 1

Set bit 6(NOIMUDAP) of IMODES33 = 1

Set bit 4(No Attitude) of DSPTAB+11 = 0, and flag for output at next opportunity

Set bit 5(Zero IMU CDU) of channel 12 = 1

CDU̲ = 0

Call "UNZ2" in 0.32 seconds

Proceed to "C33TEST"

If bit 9(IMUOPBIT) of IMODES30 = 1:

Perform "ALARM" (pattern 0213_8)

Perform "CAGESUB"

Call "ENDTNON" in 90 seconds

Proceed to "C33TEST"

<u>C33TEST</u>

TS_1 = (bits 13-11 of channel 33)

Reset channel 33 flip-flops (bits 15-11, done by a Write-type order)

If TS_1 = (bits 13-11 of IMODES33):

Proceed to "GLOCKMON"

TS = 0

Set those bits of TS = 1 that are different in IMODES33 and TS_1 (in the range of bits 13-11)

Set (bits 13-11 of IMODES33) = TS_1

If bit 13 of TS = 1: (Accelerometer fail)

Set bit 10(PIPAFLBT) of IMODES30 = bit 13(PIP2FLBT) of IMODES33
(Tag here "PIPFAIL")

Perform "SETISSW"

If bit 1(ACCFINHT) of IMODES30 = 1:

If bits 10-7 (PIPAFLBT, IMUOPBIT, ITNON2BT, ITNON1BT) and bit 5(NOACCALM) of IMODES30 all = 0:

Perform "ALARM" (pattern 0212_8)

If bit 12 of TS = 1: (Downlink too fast)

If bit 12(DNLKFAIL) of IMODES33 = 0: (Tag here "DNTMFAST")

Perform "ALARM" (pattern 1105_8)

If bit 11 of TS = 1: (Uplink too fast)

If bit 11(UPLKFAIL) of IMODES33 = 0: (Tag here "UPTMFAST")

Perform "ALARM" (pattern 1106_8)

Proceed to "GLOCKMON"

<u>GLOCKMON</u> Entered every 0.48 second after computations of "IMUMON"

$TS = |CDU_z| + K_{m70degs}$

If $TS \leq 0$:

If bit 6(Gimbal Lock) of DSPTAB+11 = 0, Resume

If bit 1(LMPTSTBT) of IMODES33 = 0:

Set bit 6(Gimbal Lock) of DSPTAB+11 = 0, and flag for output at next opportunity

Resume

If $(TS + K_{m15degs}) > 0$:

If bit 4(Coarse Align) of channel 12 = 0:

If bits 14-13 of DAPDATR1 = 11_2: (Saturn DAP)

If bit 1(AVEGFLAG) of FLAGWRD1 = 1:

Skip next 2 lines

Perform "SETCOARS"

Call "CA+ECE" in 0.06 seconds

If bit 6(Gimbal Lock) of DSPTAB+11 = 1, Resume

If bit 6(IMUNITBT) of IMODES30 = 1, Resume

Set bit 6(Gimbal Lock) of DSPTAB+11 = 1, and flag for output at next opportunity

Resume

SETISSW

TS_3 = (bits 4(IMUFINHT), 3(ICDUINHT) and 1(ACCFINHT) of IMODES30)

Shift TS_3 left 9 places (loading bits 13, 12, and 10)

TS_4 = TS_3 or IMODES30 (bits 13, 12, and 10 of IMODES30 are IMUFLBIT, ICDUFLBT, and PIPAFLBT)

TS_5 = bits 13, 12, and 10 of (- TS_4)

If $TS_5 \neq 0$: (means that 13/4, 12/3, and 10/1 bit pairs not have at least one of the two bits of the pair a binary 1)

TS = TS_5 - 1 (communication cell with "VARALARM", not affecting the TS of e.g. "IMUMON")

Perform "VARALARM"

Set bit 1(ISS Warning) of channel 11 = 1

Return

If bit 1(LMPTSTBT) of IMODES33 = 0:

Set bit 1(ISS Warning) of channel 11 = 0

Return

CAGESUB

Set bit 15(ISS Turn-on Delay Complete) and bit 6(Enable IMU CDU Error Counters) of channel 12 = 0

Set bit 5(Zero IMU CDU) and bit 4(Coarse Align) of channel 12 = 1

Set bit 4(No Attitude) of DSPTAB+11 = 1, and flag for output at next opportunity (Tag here "CAGESUB1")

Set bits 6-3 (IMUNITBT, NOACCALM, IMUFINHT, ICDUINHT) and bit 1(ACCFINHT) of IMODES30 = 1

Set bit 6(NOIMUDAP) of IMODES33 = 1

Return

ENDTNON

If bit 2(DLAYFAIL) of IMODES30 = 1:

Set bit 2(DLAYFAIL) of IMODES30 = 0

If bit 14(TONISSBT) of IMODES30 = 0:

Delay 90 seconds

Proceed to "ENDTNON"

If bit 8(IMUSE) of FLAGWRD0 = 1:

Proceed to "IMUBAD"

End of task

Set bit 15(ISS Turn-on Delay Complete) of channel 12 = 1

Set bit 4(No Attitude) of DSPTAB+11 = 0, and flag for output at next opportunity

Proceed to "UNZ2"

UNZ2

CDU = 0

Set bit 5(Zero IMU CDU) and bit 4(Coarse Align) of channel 12 = 0

Delay 7.9 seconds

Set bits 6(IMUNITBT), 4(IMUFINHT), and 3(ICDUINHT) of IMODES30 = 0

Set bit 6(NOIMUDAP) of IMODES33 = 0

Perform "SETISSW"

Set bit 15(ISS Turn-on Delay Complete) of channel 12 = 0

Call "PFAILOK" in 4 seconds

End of task

PFAILOK Called by "UNZ2"

If bit 6(IMUNITBT) of IMODES30 = 1, End of task

Set bit 10(PIPAFLBT) of IMODES30 = 1

Set bit 13(PIP2FLBT) of IMODES33 = 1

Set bit 5(NOACCALM) of IMODES30 = 0

Perform "SETISSW"

End of task

IFAILOK Called by "IMUFINE"

If bit 6(IMUNITBT) of IMODES30 = 1, End of task

If bit 4(Coarse Align) of channel 12 = 1, End of task

Set bit 13(IMUFLBIT) of IMODES30 = 1

Set bit 4(IMUFINHT) of IMODES30 = 0

Perform "SETISSW"

End of task

IMUZERO Entered from "GEOIMUTT", "GTSCPSS", and "VBZERO"

Inhibit interrupts

If bit 6(Gimbal Lock) and bit 4(No Attitude) of DSPTAB+11 = 11_2:

Perform "ALARM" (pattern 0206_8)

IMUCADR = -0

Release interrupts

Return

If bit 6(IMUNITBT) of IMODES30 = 1:

IMUCADR = -0

Release interrupts

Return

Set bits 6-5 (NOIMUDAP, IMUZROBT) of IMODES33 = 1

Set bits 4-3 (IMUFINHT, ICDUINHT) of IMODES30 = 1

Set bit 6(Enable IMU CDU Error Counters) and bit 4(Coarse Align) of channel 12 = 0

Set bit 4(No Attitude) of DSPTAB+11 = 0, and flag for output at next opportunity

Set bit 5(Zero IMU CDU) of channel 12 = 1

CDU = 0

Call "IMUZERO2" in 0.32 seconds

If bit 9(IMUOPBIT) of IMODES30 = 1:

    Perform "ALARM" (pattern 0210_8)

Release interrupts

Return

IMUZERO2

If bit 6(IMUNITBT) of IMODES30 = 1:

    Proceed to "IMUBAD"

CDU = 0

Set bit 5(Zero IMU CDU) of channel 12 = 0

Delay 7.9 seconds

If bit 6(IMUNITBT) of IMODES30 = 1:

    Proceed to "IMUBAD"

Set bits 4-3 (IMUFINHT, ICDUINHT) of IMODES30 = 0

Set bits 6-5 (NOIMUDAP, IMUZROBT) of IMODES33 = 0

Perform "SETISSW"

Proceed to "ENDIMU"

IMUCOARS

Inhibit interrupts

If bit 6(IMUNITBT) of IMODES30 = 1:

    IMUCADR = -0

    Release interrupts

    Return

Perform "SETCOARS"

Call "COARS" in 0.06 seconds

Release interrupts

Return

COARS

If bit 6(IMUNITBT) of IMODES30 = 1:

Proceed to "IMUBAD"

Set bit 6(Enable IMU CDU Error Counters) of channel 12 = 1

$\underline{COMMAND} = \underline{THETAD} - \underline{CDU}$ ones complement difference formed, and rounded shift used to rescale to B1 rev.

Delay 0.02 seconds

Proceed to "COARS2"

COARS2

If bit 6(IMUNITBT) of IMODES30 = 1:

Proceed to "IMUBAD"

$TS_1 = 0$ (in ITEMP1 cell)

Perform the following for i = Z,Y,X:

If $|COMMANDi| = 0$:

CDUiCMD = -0

If $|COMMANDi| \neq 0$:

$TS_1 = TS_1 + 1$

$TS = |COMMANDi| + K_{mcommax}$

If $TS > 0$:

COMMANDi = TS sgn COMMANDi

$CDUiCMD = - K_{mcmxm}$ sgn COMMANDi

If $TS \leq 0$:

CDUiCMD = COMMANDi

COMMANDi = 0

If $TS_1 > 0$: (i.e. non-zero commands to be sent)

Set bits 15-13(Gate outputs from $\underline{CDUXCMD}$) of channel 14 = 1

Delay 0.6 seconds

Proceed to "COARS2"

Delay 1.5 seconds

If any($|CDU_i - THETAD_i| + K_{corst}) > 0$:   ($i = x,y,z$)

Perform "ALARM" (pattern 0211_8)

Proceed to "IMUBAD"

Proceed to "ENDIMU"

CA+ECE

Set bit 6(Enable IMU CDU Error Counters) of channel 12 = 1

End of task

SETCOARS

If bit 4(Coarse Align) of channel 12 = 1, Return

Set bit 6(Enable IMU CDU Error Counters) of channel 12 = 0

Set bit 10(Gyro Output Drive) of channel 14 = 0

GYROCMD = -0

Set bit 4(Coarse Align) of channel 12 = 1

Set bit 4(No Attitude) of DSPTAB+11 = 1, and flag for output at next opportunity

Set bit 6(NOIMUDAP) of IMODES33 = 1

Set bit 4(IMUFINHT) of IMODES30 = 1

Set bits 5(TRACKFLG) of FLAGWRD1, 15(DRIFTFLG) of FLAGWRD2, and 13(REFSMFLG) of FLAGWRD3 = 0

Return

IMUFINE   Tag also "IMUFIN20"

Inhibit interrupts

If bit 6(IMUNITBT) of IMODES30 = 1:

IMUCADR = -0

Release interrupts

Return

Set bits 5(Zero IMU CDU) and 4(Coarse Align) of channel 12 = 0

Set bit 6(NOIMUDAP) of IMODES33 = 0

Set bit 4(No Attitude) of DSPTAB+11 = 0, and flag for output at next opportunity

Call "IFAILOK" in 5.12 seconds

Call "IMUFINED" in 2 seconds

Release interrupts

Return

<u>IMUFINED</u>

If bit 6(IMUNITBT) of IMODES30 = 1:

Proceed to "IMUBAD"

Proceed to "ENDIMU"

<u>PIPUSE</u> Entered from "ESTIMS" and "LASTBIAS"

PIPA = -0

If bit 6(IMUNITBT) of IMODES30 = 1, Return

Inhibit interrupts

Set bit 1(ACCFINHT) of IMODES30 = 0

Perform "SETISSW"

Release interrupts

Return

<u>PIPFREE</u> Entered from "AVGEND"

Inhibit interrupts

Set bit 1(ACCFINHT) of IMODES30 = 1

If bit 10(PIPAFLBT) of IMODES30 = 1:

Release interrupts

Return

Perform "ALARM" (pattern 0212_8)

Inhibit interrupts

Perform "SETISSW"

Release interrupts

Return

IMUPULSE

If bit 6(IMUNITBT) of IMODES30 = 1:

IMUCADR = -0

Return

If LGYRO = 0:

Set bit 6(Enable Gyro Power Supply) of channel 14 = 1

Call "STRTGYRO" in 0.04 seconds

If LGYRO $>$ 0:

Put present job to sleep (starting address id = "GWAKE")

When awakened, put to sleep again if LGYRO $>$ 0

Call "STRTGYRO" in 0.01 seconds

LGYRO = TS (TS set on entrance to "IMUPULSE" with address of first gyro command)

Force sign agreement of $\underline{E}_{LGYRO_{dp}}$ (i.e. each double precision axis)

Return

STRTGYRO

Set bits 10-7 (Gyro select bits) of channel 14 = 0

If bit 6(IMUNITBT) of IMODES30 = 1:

Proceed to "DONTPULS"

Proceed to "STRTGYR2"

STRTGYR2

If bits 14-13 of LGYRO = 00_2: (initial condition value)

TS_3 = 2

TS_4 = Bit 8 (Gyro Y-axis selection)

Proceed to "GSELECT"

If bits 14-13 of LGYRO = 01_2:

$TS_3 = 2$

TS_4 = Bits 8-7 (Gyro Z-axis selection)

Proceed to "GSELECT"

If bits 14-13 of LGYRO = 10_2:

$TS_3 = 0$

TS_4 = Bit 7 (Gyro X-axis selection)

LGYRO = LGYRO - 4 (sets for address of X component)

Proceed to "GSELECT"

LGYRO = 0 (bits 14-13 of LGYRO = 11_2)

Awaken job (if any) put to sleep with starting address id = "GWAKE"

Proceed to "IMUFINED"

<u>GSELECT</u>

LGYRO = LGYRO + TS_3 + 10000_8 (octal number is bit 13, to cause y,z,x output sequence)

ADRLGYR = (bits 11-1 of LGYRO)

$TS = E_{ADRLGYR_{dp}}$

If $(|TS| + K_{mgyrmn}) \leq 0$:

Proceed to "STRTGYR2"

$TS = TS + K_{gyrfrc}$ sgn TS (modulo 1 revolution or 2^{21} pulses)

If $TS < 0$:

$TS_4 = TS_4$ + Bit 9 (Negative gyro torquing)

Set those bits of channel 14 = 1 that are 1 in TS_4 (selects proper axis and sign)

$E_{ADRLGYR+1}$ = (bits 7-1 of $|E_{ADRLGYR+1}|$) sgn $E_{ADRLGYR_{dp}}$ (save fractional part)

$TS = K_{gyrsc} |TS|$

TS_1 = integral part of (TS/8192)

$TS_2 = TS - 8192\ TS_1$ (in range 0 - 8191)

If $TS_1 > 1$:

$E_{ADRLGYR+0} = TS_1 - 2$ (Tag here "LONGGYRO")

$GYROCMD = 8192 + TS_2$

Call "8192AUG" in (K_{bt10} GYROCMD - 3) centi-seconds

Set bit 10(Gyro Output Drive) of channel 14 = 1

End of task

If $TS_1 = 1$:

$TS_2 = TS_2 + 8192$

$E_{ADRLGYR+0} = 0$

$GYROCMD = TS_2$

If bit 4(IMU Coarse Align) of channel 12 = 0:

If bit 6(IMUNITBT) of IMODES30 = 0:

Call "TWOPULSE" in (K_{bt10} GYROCMD + 3) centi-seconds

Set bit 10(Gyro Output Drive) of channel 14 = 1

End of task

Proceed to "DONTPULS"

<u>8192AUG</u>

If bit 4(IMU Coarse Align) of channel 12 = 1:

Proceed to "DONTPULS"

If bit 6(IMUNITBT) of IMODES30 = 1:

Proceed to "DONTPULS"

If $E_{ADRLGYR+0} > 0$:

$E_{ADRLGYR+0} = E_{ADRLGYR+0} - 1$

GYROCMD = GYROCMD + 8192

Call "8192AUG" in (K_{bt10} GYROCMD - 3) centi-seconds

Set bit 10(Gyro Output Drive) of channel 14 = 1

End of task

GYROCMD = GYROCMD + 8192

If bit 4(IMU Coarse Align) of channel 12 = 0: (redundant with check already done)

If bit 6(IMUNITBT) of IMODES30 = 0:

Call "TWOPULSE" in (K_{bt10} GYROCMD + 3) centi-seconds

Set bit 10(Gyro Output Drive) of channel 14 = 1

End of task

Proceed to "DONTPULS"

TWOPULSE (Purpose is to have "last" torquing always negative)

Set bit 9(Negative Gyro Torquing) of channel 14 = 0

GYROCMD = 2 (sent in less than 1 ms)

Set bit 10(Gyro Output Drive) of channel 14 = 1

Call "STRTGYRO" in 0.01 second (logic for this requires over 1 ms)

Set bit 9(Negative Gyro Torquing) of channel 14 = 1

GYROCMD = 2

Set bit 10(Gyro Output Drive) of channel 14 = 1

End of task

DONTPULS

LGYRO = 0

Awaken job (if any) put to sleep with starting address id = "GWAKE"

Proceed to "IMUBAD"

ENDIMU

If bit 1(ISS Warning) of channel 11 = 1:

Proceed to "IMUBAD"

If IMUCADR = +0:

IMUCADR = -1

End of task

Awaken job put to sleep in "IMUSTALL" and cause it to start at (IMUCADR +1), indicating a success

IMUCADR = +0

End of task

IMUBAD

If IMUCADR = +0:

IMUCADR = -0

End of task

Awaken job put to sleep in "IMUSTALL" and cause it to start at (IMUCADR), indicating an error

IMUCADR = +0

End of task

IMUSTALL

If IMUCADR > 0, proceed to "POODOO" (pattern 21210_8)

If IMUCADR = +0, set IMUCADR = (Calling address +1, in FCADR format), and put present job to sleep

If IMUCADR = -0:

IMUCADR = +0

Return (to calling address +1, indicating an error)

If IMUCADR = -1:

IMUCADR = +0

Return (to calling address +2, indicating a success)

If IMUCADR < -1, proceed to "POODOO" (pattern 21210_8)

Quantities in Computations

See also list of major variables and list of routines

1dPIPADT: Cell containing time information used with IMU compensation routines. During free-flight portions, contains value of TIME1 when previous gyro drift compensation was made, scale factor B14, units centi-seconds (with optional tag "OLDBT1"). During those portions of flight when "1/PIPA" is entered, contains value of computing interval, scale factor B8, units centi-seconds. Both quantities are single precision. Cell is set to 2 seconds in "P11" and in "LASTBIAS", and to $\frac{1}{2}$ second in "GTSCPSS".

ADRLGYR: Dummy quantity used to indicate that gyro compensation information taken from least significant 11 bits of LGYRO. Bits 11-9 are loaded into EBANK (see Data Input/Output) and bits 8-1 are added to 1400_8 to give address within the erasable memory bank.

$\underline{C}_{ad}$: Set of three single precision erasable memory constants, program notation "ADIAX", "ADIAY", and "ADIAZ", to correct for "IRIG acceleration sensitive drift along the input axis", scale factor B-3, units gyro pulses/accelerometer count. One gyro pulse is 2^{-21} revolution and one accelerometer count is a velocity increment of 5.85 centimeters/second. The units of the constant can also be expressed as "meru/g", where one meru is 10^{-3} x earth rate. In these units, the scaling of the constant is about 19.0304345 least increments per (meru/g): this figure is 2^{17} x 0.024339048 x (5.85 x 10^{-2})/9.80665, where first term is the reciprocal of the least increment for units of gyro pulses/accelerometer count; the second term is the number of gyro pulses/meru (0.1 K_{omegms}, see Prelaunch Alignment); the third is the accelerometer scale factor in units of meters/second; and the fourth is gravity (meters/sec^2). If earth rate is simplified to 15 degrees/hour, the number of least increments per (meru/g) is reduced to about 18.9785.

$\underline{C}_{nbd}$: Set of three single precision erasable memory constants, program notation "NBDX", "NBDY", and "NBDZ", to correct for "IRIG bias drift", scale factor B-5, units gyro pulses/centi-second. The units of the constant can also be expressed as "meru", in which case the scaling is about 127.606708 least increments per meru: this figure is 2^{19} x 0.01 x 0.024339048 (see $\underline{C}_{ad}$: the 0.01 factor is for centi-seconds). For earth rate simplified to 15 degrees/hour, the number of least increments per meru is reduced to about 127.2583.

$\underline{C}_{pipabias}$: Set of three single precision erasable memory constants, program notation for first "PIPABIAS" (or "PBIASX", "PBIASY", and "PBIASZ"), to correct for "accelerometer bias", scale factor B-6, units accelerometer counts/centi-second. The units of the constant can also be expressed as "cm/sec^2", in which case the scaling is about 1792.4376 least increments per cm/sec^2: this figure is 2^{20} x 0.01 x (1/5.85), where first term is reciprocal of normal least increment, second compensates for centi-seconds,

and third is reciprocal of accelerometer scale factor. In-flight evaluation could be performed by determining the number of accelerometer counts accumulated in a period of 320 seconds, multiplying by two to give counts in (2^6 x 10) seconds, and then dividing by 1000 to give scaled counts/centi-second. N21 can be used to read PIPA cells and N01 with decimal input can be used to load the appropriate cells (1452, 1454, and 1456 in ECADR form for x, y, z respectively).

$\underline{C}_{pipascf}$: Set of three single precision erasable memory constants, program notation "PIPASCFX", "PIPASCFY", and "PIPASCFZ", scale factor B-9, units accelerometer counts/accelerometer count, used to correct for "accelerometer scale factor error." The units of the constant can also be expressed as "parts per million (PPM)", in which case the scaling is 8.388608 least increments per PPM: this figure is 2^{23} x 10^{-6}, where first term is reciprocal of normal least increment and second is PPM conversion.

$\underline{C}_{sr}$: Set of three single precision erasable memory constants, program notation "ADSRAX", "ADSRAY", and "ADSRAZ", to correct for "IRIG acceleration sensitive drift along the spin reference axis", scale factor B-3, units gyro pulses/accelerometer count. See discussion with $\underline{C}_{ad}$ (which has same scaling).

CDUXCMD (CDUXCMD, CDUYCMD, CDUZCMD): Single precision value of computer special erasable memory cells 0050_8 - 0052_8. Output pulses are generated based on the contents of these cells if bits 15-13 of channel 14 (respectively) are set 1, and the respective IMU CDU error counter is loaded from the pulse train information if bit 6 of channel 12 is 1. Pulses may be used to coarse align the IMU if bit 4 of channel 12 is set (in this case, IMU stable member movement causes the error counter information to be decremented), with 2^{13} pulses giving one revolution. Pulses are generated at a rate of 3200 pps. See also Digital Autopilot Interface Routines.

COMMAND: Single precision value of required coarse-align command, scale factor B1, units revolutions. Could also be considered (see above discussion of CDUiCMD) to have a scale factor B14, units coarse-align pulses.

DAPDATR1: See Digital Autopilot Interface Routines.

GCOMP: Value of required gyro compensation command, computed with a scale factor B14 but used in "IMUPULSE" with a scale factor B21 (or, alternatively, with a scale factor B0 revolutions rather than B21 gyro pulses, since there are 2^{21} gyro pulses/revolution).

GCOMPSW: Single precision control cell used to bypass the performance of "1/PIPA" and "NBDONLY" if is negative. Set 0 in "GTSCPSS".

GYROCMD: Single precision value of computer special erasable memory cell 0047_8, used to control generation of gyro torquing pulses at a 3200 pps rate if bit 10 of channel 14 = 1. Gyro power supply for the pulses is enabled by bit 6 of channel 14, bits 8-7 indicate the axis, and bit 9 is 1 if a negative command is to be produced. One pulse is 2^{-21} revolution.

IMODES30: Single precision flag word used for control of "T4RUPT" routines associated with channel 30 inputs. Individual bits have the meanings given below. A restart ("GOPROG") sets bits 14-10 to 1 and sets bits 15, 8-6, and 2 to 0.

Bit	Symbol	Meaning
15	TLIMBIT	Last sampled value of channel 30 bit 15 (0 if stable member within designed temperature limits).
14	TONISSBT	Last sampled value of channel 30 bit 14 (0 if ISS has been turned on or commanded to be turned on).
13	IMUFLBIT	Last sampled value of channel 30 bit 13 (0 if an IMU fail indication has been produced by the IMU hardware).
12	ICDUFLBT	Last sampled value of channel 30 bit 12 (0 if an IMU CDU fail indication has been generated by the IMU CDU hardware).
11	CAGEBIT	Last sampled value of channel 30 bit 11 (0 if an IMU cage command generated by the crew).
10	PIPAFLBT	Same as bit 13 of IMODES33 (0 if an accelerometer fail indication produced by the hardware, channel 33 bit 13).
9	IMUOPBIT	Last sampled value of channel 30 bit 9 (0 if IMU turned on and operating with no malfunctions).
8	ITNON2BT	Bit 7 of IMODES30 sensed as 1 (i.e. turn-on request received). Used to achieve a wait of 0.48 seconds (during which bit is 1) before acting on turn-on information.
7	ITNON1BT	IMU turn-on request received (a change in value to a logic 1, i.e. binary 0, of bit 14 or bit 9 of IMODES30).
6	IMUNITBT	Bit set 1 to indicate that IMU use should not be attempted: set 1 during the turn-on sequence, if bit 11 of IMODES30 indicates a cage command, or ISS zeroing in "T4RUPT" loop.
5	NOACCALM	Bit set 1 to inhibit generation of program alarm 212_8 in "C33TEST" (for accelerometer fail signal).
4	IMUFINHT	Bit set 1 to inhibit generation of ISS Warning signal based on IMU fail signal (bit 13 of IMODES30).

Bit	Symbol	Meaning
3	ICDUINHT	Bit set 1 to inhibit generation of ISS Warning signal based on receipt of IMU CDU fail signal (bit 12 of IMODES30).
2	DLAYFAIL	Bit set 1 to indicate failure of the turn-on delay sequence (IMU turn-on signals not present for the required time interval).
1	ACCFINHT	Bit set 1 to inhibit generation of ISS Warning signal based on receipt of accelerometer fail signal (bit 10 of IMODES30).

IMODES33: Single precision flag word used for control of "T4RUPT" routines associated with channel 33 inputs (and other functions). Individual bits have the following meanings.

Bit	Symbol	Meaning
14	PROCDBIT	Last sampled value of channel 32 bit 14 (0 if a Proceed pushbutton command given). This bit is used in "PROCEEDE".
13	PIP2FLBT	Last sampled value of channel 33 bit 13 (0 if an accelerometer fail indication has been produced by the hardware).
12	DNLKFAIL	Last sampled value of channel 33 bit 12 (0 if telemetry end pulse rejected, downlink too fast).
11	UPLKFAIL	Last sampled value of channel 33 bit 11 (0 if uplink pulse rejected, uplink too fast).
6	NOIMUDAP	Bit set 1 to indicate that IMU use for vehicle attitude information should not be attempted (sensed by RCS DAP and Entry DAP).
5	IMUZROBT	Bit set 1 in "IMUZERO" and reset 0 about 8.22 seconds later in "IMUZERO2". Indicates that IMU zeroing outside of T4RUPT package is taking place.
1	LMPTSTBT	Bit set 1 if a lamp test (verb 35) is in progress, to inhibit turn-off of appropriate indicators.

IMUCADR: Cell containing control information used in association with "IMUSTALL": it contains return address information from the routine or information on the success (if = -1) or failure (if = -0) of an IMU function.

K_{2secp}: Single precision constant, program notation "PRIO31", octal value 31000_8, scale factor B8, units centi-seconds. Value corresponds to 2 seconds.

K_{bt10}: Single precision constant, program notation "BIT10", scale factor B0, units centi-seconds/gyro torquing pulse. Value is 2^{-5}(i.e. 1/32), since output pulse rate is 3200 pps, or 32 pulses/centi-second.

K_{corst}: Single precision constant, program notation "COARSTOL", scale factor B-1, units revolutions. Octal value is 77511_8, which corresponds (after a one-bit modification used in the program for convenience in forming the absolute value) to about -2.0105°.

K_{cpck}: Single precision constant, program notation "NEGONE", scale factor B14, units gyro pulses, used to check if compensation of gyros is required. Octal value is 77776_8, but used in the program in such a way (a mask on decremented $|GCOMP_i|$) that effective value corresponds to 3 pulses: the most significant half of $GCOMP_i$ must be 3 or more pulses for the "1/GYRO" computations to be initiated, although lack of sign agreement could make the actual value only slightly more than 2 pulses.

K_{gyrfrc}: Constant, program notation "GYROFRAC", scale factor B21, units gyro pulses. Stored value is 28×2^{-28}, corresponding to about 0.219 pulses (nominal value 0.215).

K_{gyrsc}: Dummy constant indicating a rescaling of original contents of $E_{ADRLGYR}$ from B21 to B28 pulses. "Value" is 1.0, scale factor B7.

$K_{m15degs}$: Single precision constant, program notation "-15DEGS", scale factor B-1, units revolutions. Octal value is 75252_8, corresponding to about -14.996°.

$K_{m70degs}$: Single precision constant, program notation "-70DEGS", scale factor B-1, units revolutions. Octal value is 63434_8, corresponding to about -70.005° (after a one-bit modification used in the program for convenience in forming the absolute value).

K_{mcmxm}: Single precision constant, program notation "-COMMAX-", scale factor B14, units CDU output pulses. Value is -192×2^{-14}.

$K_{mcommax}$: Single precision constant, program notation "-COMMAX", scale factor B14, units CDU output pulses. Value is -191×2^{-14}, but used in program (for convenience in forming the absolute value) in such a way that effective value is -192. Limits change in CDU angle to 192 least increments per axis (or 8.4375°) each 0.6 seconds. The maximum pulses required (to go 180 degrees) would be 4096, which would require "21.3333", i.e. 22 intervals of 0.6 seconds each. With the additional 1.5 seconds for the K_{corst} check, time required for coarse align would be less than about 0.06 + 0.02 + 22 (0.6) + 1.5, or 14.78 seconds.

K_{mgyrmn}: Single precision constant, program notation "-GYROMIN", scale factor B7, units gyro pulses. Octal value is 77601_8, which corresponds (after a one-bit adjustment used in program for convenience in forming the absolute value) to (-127/128), i.e. one least increment less than 1.00 gyro pulse.

LGYRO: Cell containing (if non-zero) address information for gyro torquing commands. If cell is zero, gyros are "available" (i.e. not being driven).

PIPA: Value of special erasable memory cells (0037_8 - 0041_8) containing accumulated output counts from the accelerometers, scale factor B14, units counts. One count is 5.85 centimeters/second.

T_{pptm1}: Value of computer clock when accelerometers sampled, program notation "PIPTIME1", scale factor B28, units centi-seconds. Subsequently loaded into T_{pptm}, hence can also be used as storage for the time tag associated with $\underline{R}_{n1}$ and $\underline{V}_{n1}$ (see General Program Control) for use by "PREREAD".

Inflight Alignment

P51 (Tag also "P53")

If bit 9(IMUOPBIT) of IMODES30 = 1: (IMU not on)

Perform "ALARM" (pattern 0210_8)

Proceed to "GOTOPOOH"

Proceed to "P51A"

P51A

Set bit 8(IMUSE) of FLAGWRD0 = 1

TS = 00015_8

Proceed to "GOPERF1": if terminate, proceed to "GOTOPOOH"
if proceed, proceed to "P51B"
otherwise, proceed

THETA$\underline{D}$ = 0

TS = 0622_{vn}

Perform "GODSPRET"

TS = 4100_{vn}

Perform "GODSPRET"

If IMUCADR $\neq$ 0:

Delay 1 second (by putting job to sleep via "DELAYJOB")

Proceed to second previous line (recheck IMUCADR)

Perform "IMUCOARS"

Perform "IMUSTALL": if error return, perform "217ALARM"; proceed
otherwise, proceed

Perform "IMUFINE"

Perform "IMUSTALL": if error return, perform "217ALARM"; proceed
otherwise, proceed

Proceed to second line of "P51A"

P51B

STARIND = 0

ldPIPADT = TIME1

GCOMP = 0

Set bit 15(DRIFTFLG) of FLAGWRD2 = 1

Set bit 9(LMKTRG) of FLAGWRD1 = 0

If MODREG = 53: (Tag here "P51C", from below)

    Perform "R56"

If MODREG ≠ 53:

    Perform "R53"

Perform "SXTSM"

If STARIND = 0:

    STARSAV1 = TS

    TS = T_{sight}

    Perform "PLANET"

    PLANVEC = TS

    STARIND = 1

    Proceed to 6th line of "P51B"

STARSAV2 = TS

TS = T_{sight} (Tag here "P51F")

Perform "PLANET"

STARBC = TS

STARAC = PLANVEC

STARAD = STARSAV1

STARBD = STARSAV2

Perform "R54"

If bit 3(FREEFLAG) of FLAGWRD0 = 0: (e.g. a V32E response in "R54")

Proceed to second line of "P51A"

Perform "AXISGEN"

Set bit 13(REFSMFLG) of FLAGWRD3 = 0

$[REFSMMAT] = [X_{dc}]$

Set bit 13(REFSMFLG) of FLAGWRD3 = 1

Proceed to "GOTOPOOH"

<u>PROG52</u> (Tag also "P54")

Set bits 7(UPDATFLG) and 5(TRACKFLG) of FLAGWRD1 = 0

Perform "R02BOTH"

If bit 7(AUTOSEQ) of FLGWRD10 = 1:

Proceed to "P52AUTO" (page MINK-9)

If bit 9(UTFLAG) of FLAGWRD8 = 1:

If OPTNTYPE ≠ 0: (means "option 2 of P20", i.e. R67 rotation)

Set bit 5(TRACKFLG) of FLAGWRD1 = 1 (should <u>not</u> have R67 active in P54 because of delay it can introduce in "R56")

If bit 4(PFRATFLG) of FLAGWRD2 = 0:

OPTION2 = 3

If bit 4(PFRATFLG) of FLAGWRD2 = 1:

OPTION2 = 1

Proceed to "P52B"

<u>P52B</u>

TS = 1

Proceed to "GOPERF4": if terminate, proceed to "GOTOPOOH"
if proceed, proceed
otherwise, proceed to previous line

TS = (bits 2-1 of OPTION2)

If TS = 3: ([REFSMMAT] orientation)

Proceed to "P52C"

If TS = 1: ("preferred" orientation)

Proceed to "P52D"

If TS = 0 or 2, proceed ("nominal" or "landing site" orientations)

$DSPTEM1_{dp}$ = (-0, 0)

TS = 0634_{vn}

Proceed to "VNFLASH": (if terminate, proceed to "GOTOPOOH")
if proceed, proceed
otherwise, proceed to previous line

If $DSPTEM1_{sp}$ = 0:

$DSPTEM1_{dp} = T_{now}$

If bit 2 of OPTION2 = 1: ("nominal" orientation)

$T_{decl} = DSPTEM1_{dp}$ (Tag here "S52.3")

Perform "CSMPREC"

$\underline{Z}_{smd} = -\ \text{unit}\underline{R}_{att}$

$\underline{Y}_{smd} = \text{unit}(\underline{V}_{att} * \underline{R}_{att})$

$\underline{X}_{smd} = \text{unit}(\underline{Y}_{smd} * \underline{Z}_{smd})$

Proceed to "P52D"

Set bit 12(LUNLATLO) of FLAGWRD3 = 1 ("landing site" orientation)

Set bit 13(ERADCOMP) of FLAGWRD1 = 1 (mean lunar radius altitude base)

$T\underline{S}_1 = RL\underline{S}$

$T_{sight} = DSPTEM1_{dp}$

$TS_2 = DSPTEM1_{dp}$

TS = $DSPTEM1_{dp}$ (non-zero, meaning moon)

Perform "RP-TO-R"

ALPHA$\underline{V}$ = T$\underline{S}$, shifted right 2 places (to scale factor B29)

$TS = T_{sight}$

Perform "LAT-LONG"

$LANDLONG = \frac{1}{2}\ LONG$

LANDALT = ALT

$TS = 0689_{vn}$

Proceed to "VNFLASH": (if terminate, proceed to "GOTOPOOH")
if proceed, proceed
otherwise, proceed to previous line

LONG = 2 LANDLONG (note that the checks of "P22SUBRB" not done here)

ALT = LANDALT

$TS = T_{sight}$

Perform "LALOTORV"

$\underline{X}_{smd} = \text{unit}\underline{\text{ALPHAV}}$

$T_{decl} = T_{sight}$

Perform "CSMPREC"

$\underline{Z}_{smd} = \text{unit}\left((\underline{R}_{att} * \underline{V}_{att}) * \underline{X}_{smd} \right)$

$\underline{Y}_{smd} = \text{unit}(\underline{Z}_{smd} * \underline{X}_{smd})$

Proceed to "P52D"

P52D

Perform "S52.2"

$TS = 0622_{vn}$

Proceed to "GOFLASH": if terminate, proceed to "GOTOPOOH"
if proceed, proceed
otherwise, proceed to "P52D"

If bit 7(AUTOSEQ) of FLGWRD10 = 1:

Proceed to "PERF20" (page MINK-9)

$TS = 00013_8$

Proceed to "GOPERF1": if terminate, proceed to "GOTOPOOH"
if proceed, proceed
otherwise, proceed to "GYCRS"

Proceed to "CAL53A" (exits to "P52C")

P52C

$TS = 00015_8$

Proceed to "GOPERF1": if terminate, proceed to "GOTOPOOH"
if proceed, proceed
otherwise, proceed to "R51"

$T_{sight} = T_{now} + K_{tsgtl}$

Perform "S50"

Perform "CDUTRIG" (This is start of the former "PICAPAR" routine: the tag has been deleted)

Perform "CALCSMSC"

Set bit 10(VFLAG) of FLAGWRD3 = 1 (means no star found)

BESTI = 0

BESTJ = 0

$SA\underline{X} = \text{unit}\left((K_{sin33}\ \underline{X}_{dc} + K_{cos33}\ \underline{Z}_{dc}) \left[\text{REFSMMAT}\right]\right)$

X1 = 228

Proceed to "PIC1"

PIC1

If X1 $\leq$ 6, proceed to "PICEND"

X1 = X1 - 6

$T\underline{S} = \underline{K}_{cat_{X1}}$

Perform "OCCULT"

If bit 7(CULTFLAG) of FLAGWRD3 = 1, proceed to "PIC1"

X2 = X1

Proceed to "PIC3"

PIC3

If X2 $\leq$ 6, proceed to "PIC1"

X2 = X2 - 6

$T\underline{S} = \underline{K}_{cat_{X2}}$

Perform "OCCULT"

If bit 7(CULTFLAG) of FLAGWRD3 = 1, proceed to "PIC3"

$TS = \underline{K}_{cat_{X1}} \cdot \underline{K}_{cat_{X2}} - K_{css66}$

If TS $<$ 0, proceed to "PIC3" (separation more than 76^o)

If TS + $K_{css6640} \geq$ 0, proceed to "PIC3" (separation 30^o or less)

If $\underline{K}_{cat_{X1}} \cdot \underline{SAX} - K_{css33} < 0$, proceed to "PIC1"

If $\underline{K}_{cat_{X2}} \cdot \underline{SAX} - K_{css33} < 0$, proceed to "PIC3"

If bit 10(VFLAG) of FLAGWRD3 = 1: (means no star yet found)

Set bit 10(VFLAG) of FLAGWRD3 = 0

BESTI = X1

BESTJ = X2

Proceed to "PIC3"

$TS_1 = \underline{K}_{cat_{BESTI}} \cdot \underline{K}_{cat_{BESTJ}}$

Set bit 10(VFLAG) of FLAGWRD3 = 1

$TS_2 = \underline{K}_{cat_{X1}} \cdot \underline{K}_{cat_{X2}}$

Set bit 10(VFLAG) of FLAGWRD3 = 0

If $TS_2 - TS_1 \geq 0$, proceed to "PIC3" (new pair closer than old pair)

BESTI = X1

BESTJ = X2

Proceed to "PIC3"

OCCULT

If CEARTH - ($T\underline{S} \cdot VEART\underline{H}$) $\leq$ 0:

Set bit 7(CULTFLAG) of FLAGWRD3 = 1

Return

If CSUN - ($T\underline{S} \cdot VSU\underline{N}$) $<$ 0:

Set bit 7(CULTFLAG) of FLAGWRD3 = 1

Return

If CMOON - ($T\underline{S} \cdot VMOO\underline{N}$) < 0:

Set bit 7(CULTFLAG) of FLAGWRD3 = 1

Return

Set bit 7(CULTFLAG) of FLAGWRD3 = 0

Return

<u>PICEND</u> Entered from "PIC1" at the end of the star table scan

If bit 10(VFLAG) of FLAGWRD3 = 1: (means no stars found)

Perform "ALARM" (pattern 0405_8)

TS = 0509_{vn}

Proceed to "GOFLASH": if terminate, proceed to "GOTOPOOH"
if proceed, proceed
otherwise, proceed to "P52C"

Proceed to "R51"

<u>S50</u> (tag also "LOCSAM")

TS = T_{sight}

Perform "LSPOS"

$VMOO\underline{N} = T\underline{S}$ (moon position, B29 meters)

$VSU\underline{N} = T\underline{S}_1$ (sun position, B38 meters)

$T_{decl} = T_{sight}$

Perform "CSMCONIC"

If X2 = 0: (means earth-centered)

$VMOO\underline{N} = \text{unit}(VMOO\underline{N} - \underline{R}_{att})$

$VEART\underline{H} = -\text{unit}\underline{R}_{att}$

$CEARTH = \cos\left(\sin^{-1}(K_{rsube} / |\underline{R}_{att}|) + K_{5degs}\right)$

$CMOON = K_{css5}$

$VSU\underline{N} = \text{unit}VSU\underline{N}$

If X2 > 0: (means moon-centered)

$VSU\underline{N} = \text{unit}(VSU\underline{N} - VMOO\underline{N})$

$VEART\underline{H} = -\text{unit}(VMOO\underline{N} + \underline{R}_{att})$

(If X2 > 0):

$\underline{VMOON} = - \text{unit}\underline{R}_{att}$

$CMOON = \cos\left(\sin^{-1}(K_{rsubm} / |\underline{R}_{att}|) + K_{5degs}\right)$

$CEARTH = K_{css5}$

$\underline{VELdC} = K_{ldc}\ \underline{V}_{att} + \underline{VSUN} * \underline{K}_{eclipol}$

$CSUN = K_{cssun}$

Return

S52.2

Perform "CDUTRIG"

Perform "CALCSMSC"

$\underline{X}_{dc} = \text{unit}(\underline{X}_{dc}\ [REFSMMAT])$

$\underline{Y}_{dc} = \text{unit}(\underline{Y}_{dc}\ [REFSMMAT])$

$\underline{Z}_{dc} = \text{unit}(\underline{Z}_{dc}\ [REFSMMAT])$

$[X_{sm}] = [X_{smd}]$

Perform "CALCGA"

Return

PLANET

$T_{sight} = TS$

Perform "S50"

Set $\underline{VSUN} = \underline{VEARTH}$ and $\underline{VEARTH} = \underline{VSUN}$ (for indexing below)

TS = 6 (bits 6-1 of STARCODE)

If STARIND = 0:

BESTI = TS

If STARIND = 1:

BESTJ = TS

If TS = 0: (means planet)

$TS = 0688_{vn}$

Proceed to "VNFLASH": (if terminate, proceed to "GOTOPOOH")
if proceed, proceed
otherwise, proceed to previous line

$T\underline{S} = \text{unit}\left(\text{unit}(K_{1dsqrt3}\ \text{STARSAV}\underline{3}) + \text{VELd}\underline{C}\right)$

Return

If TS - 228 < 0: (original STARCODE was in range $01\text{-}45_8$, meaning a star in star table)

$T\underline{S} = \text{unit}(\ \underline{K}_{cat_{TS}} + \text{VELd}\underline{C})$

Return

If TS = 228: (original STARCODE was 46_8, meaning sun)

$T\underline{S} = \text{unit}(\ \text{VEART}\underline{H} + \text{VELd}\underline{C})$ (VEART$\underline{H}$ has sun location)

Return

If TS = 234: (original STARCODE was 47_8, meaning earth)

$T\underline{S} = \text{unit}\ (\ \text{VSU}\underline{N} + \text{VELd}\underline{C})$ (VSU$\underline{N}$ has earth location)

Return

If TS = 240: (original STARCODE was 50_8, meaning moon)

$T\underline{S} = \text{unit}\ (\ \text{VMOO}\underline{N} + \text{VELd}\underline{C})$

Return

If TS > 240, meaning STARCODE in range 51_8 - 77_8 (for low six bits), improper results. N70/N71 loads check for legal range, however.

<u>SXTSM</u>

$T_{sight} = \text{MRKBUF1}_{dp}$

X1 = - "MRKBUF1"

If STARIND = 0: (Tag here "SXTSM1", entered from "GETUM")

$\text{MARKDOWN+i} = E_{i-X1}$ (i = 0 - 6) ("GETUM" point, since STARIND = 0 in "ROO")

If STARIND = 1:

$\text{MARK2DWN+i} = E_{i-X1}$ (i = 0 - 6)

$CDUSPOT = E_{2-X1}$

Perform "SXTNB"

Perform "TRG*NBSM"   (leaves line-of-sight in $\underline{TS}$)

Return

CAL53A   (This is routine R50, entered from "P52D")

Perform "S52.2"

$\underline{TS} = \underline{CDU}$

If any ($|TS_i - THETAD_i|$) $\geq K_{deg1}$ and $< K_{deg359}$:   (i = x,y,z)

If IMUCADR $\neq$ 0:

Delay 1 second (by putting job to sleep via "DELAYJOB")

Proceed to second previous line (recheck IMUCADR)

Perform "IMUCOARS"

Perform "IMUSTALL": if error return, perform "217ALARM"; proceed otherwise, proceed

Perform "IMUFINE"

Perform "IMUSTALL": if error return, perform "217ALARM"; proceed otherwise, proceed

1dPIPADT = TIME1

$\underline{GCOMP} = 0$

Set bit 15(DRIFTFLG) of FLAGWRD2 = 1

$[REFSMMAT] = [X_{sm}]$

Set bit 13(REFSMFLG) of FLAGWRD3 = 1

Proceed to "P52C"

R51   Entered from "PICEND" and "P52C"

STARIND = 1

Set bit 9(LMKTRG) of FLAGWRD1 = 0

```
If STARIND = 0:

     STARCODE =  BESTI / 6

If STARIND = 1:

     STARCODE =  BESTJ / 6

Perform "R51DSPA"

If  STARIND > 0:

     STARSAV1 = TS

If  STARIND = 0:

     STARSAV2 = TS

If (MODREG modulo 4) ≠ 0:       (i.e. P54 rather than P52)

     Perform "R56"      (processing of mark ENTR could be delayed if
                         R67 is active)

If (MODREG modulo 4) = 0:

     Perform "R52"

Perform "SXTSM"         (Tag here "R51B")

STARSAV2 = TS

TS = T_sight

Perform "PLANET"

If  STARIND > 0:

     PLANVEC =  unit( [REFSMMAT]  TS )

     STARSAV1 = STARSAV2

     STARIND = 0

     Proceed to 2nd line of "R51"

STARAD = unit( [REFSMMAT]  TS )

STARAC = STARSAV2

STARBC = STARSAV1

STARBD = PLANVEC
```

Perform "R54"

If bit 3(FREEFLAG) of FLAGWRDO = 0:

Proceed to "R51K"

Perform "AXISGEN"

Proceed to "R55"   (exits to "R51K")

R51K

$TS = 00014_8$ (Tag here also "R51KA")

Proceed to "GOPERF1": if terminate, proceed to "GOTOPOOH"
if proceed, proceed
otherwise, proceed to "GOTOPOOH"

Proceed to "P52C"

R51DSPA   Entered from "R51" and "R57"

$TS = 0170_{vn}$

Perform "VNFLASHR": (if terminate, proceed to "GOTOPOOH")
if proceed, skip next 2 lines
otherwise, proceed to previous line

$TS = 110_2$ and perform "BLANKET"   (R3BLNK, R2BLNK)

End of job

If STARCODE ≤ -0 or if STARCODE $> 50_8$:

Set bit 7(Operator Error) of channel 11 = 1

Proceed to "R51DSPA"

$TS = T_{now}$

Perform "PLANET"

Return

R54   (Tag also "CHKSDATA"; entered from "P51B" and "R51")

Set bit 3(FREEFLAG) of FLAGWRDO = 1

$TS_1 = \cos^{-1}$ ($\underline{STARAD} \cdot \underline{STARBD}$)

Set bit 3(FREEFLAG) of FLAGWRDO = 0

$TS_2 = \cos^{-1}$ ($\underline{STARAC} \cdot \underline{STARBC}$)

DSPTEM1 = $|TS_2 - TS_1|$, with sign agreement forced

Set bit 3(FREEFLAG) of FLAGWRDO = 1

TS = 0

Perform "CLEANDSP"

TS = 0605_{vn}

Proceed to "GOFLASH": if terminate, proceed to "GOTOPOOH"
if proceed, skip next line
otherwise, proceed

Set bit 3(FREEFLAG) of FLAGWRDO = 0 (e.g. a V32E response)

Return

R55 Entered from "R51"

Perform "CALCGTA"

TS = 0693_{vn}

Proceed to "GOFLASH": if terminate, proceed to "GOTOPOOH"
if proceed, proceed
otherwise, proceed to "R55RET"

TS = "OGC"

Perform "IMUPULSE"

Perform "IMUSTALL": if error return, perform "217ALARM"; proceed
otherwise, proceed

Proceed to "R55RET"

R55RET

Set bit 4(PFRATFLG) of FLAGWRD2 = 0

Proceed to "R51K"

GYCRS Entered from "PERF20" and "P52D"

$\underline{X}_{dc}$ = unit([REFSMMAT] $\underline{X}_{smd}$)

$\underline{Y}_{dc}$ = unit([REFSMMAT] $\underline{Y}_{smd}$)

$\underline{Z}_{dc}$ = unit([REFSMMAT] $\underline{Z}_{smd}$)

Perform "CALCGTA"

Set bit 15(DRIFTFLG) of FLAGWRD2 = 0

Set bit 13(REFSMFLG) of FLAGWRD3 = 0

$TS = 1620_{vn}$

Perform "GODSPR"

TS = "OGC"

Perform "IMUPULSE"

Perform "IMUSTALL": if error return, perform "217ALARM"; proceed
otherwise, proceed

$[REFSMMAT] = [X_{smd}]$

Set bit 4(PFRATFLG) of FLAGWRD2 = 0

Set bit 13(REFSMFLG) of FLAGWRD3 = 1

1dPIPADT = TIME1

GCOMP = 0

Set bit 15(DRIFTFLG) of FLAGWRD2 = 1

If bit 7(AUTOSEQ) of FLGWRD10 = 1:

Set bit 1(PCFLAG) of FLGWRD10 = 0 (means did pulse torquing)

Proceed to second line of "GOTOPOOH"

Proceed to "R51K"

217ALARM

Inhibit interrupts

ALMCADR = "Calling address +1" (S-register portion)

Perform "ALARM2" (pattern 0217_8)

Return (address in ALMCADR)

Quantities in Computations

See also list of major variables and list of routines

1dPIPADT: See IMU Computations.

ALMCADR: See General Program Control.

ALPHA$\underline{V}$: See Coordinate Transformations.

ALT: See Coordinate Transformations.

BESTI: Single precision value of index parameter for star #1 of the "best" star pair (as determined by "P52C" computations), scale factor B14. It will be zero if no star pairs found that are satisfactory, and otherwise will be equal to six times the "star number" (see $\underline{K}_{cat}$) when loaded in "P52C" computations. See BESTJ.

BESTJ: Single precision value of index parameter for star #2 of the "best" star pair (as determined by "P52C" computations), scale factor B14. It will be zero if no star pairs found that are satisfactory, and otherwise will be equal to six times the "star number" (see $\underline{K}_{cat}$) when loaded in "P52C" computations. Unless it is zero, it will be less than BESTI as loaded in "P52C" computations. In P52/P54, it is loaded with (STARCODE x 6), see "PLANET", for the first body that is used, while BESTI has the second; in P51/P53, however, it contains (STARCODE x 6) for the second body, while BESTI has the first. In either case, however, BESTJ will contain the body code associated with the sighting data for which the address selected in "SXTSM" is MARK2DWN. "BESTJ" = "BESTI" + 1.

CDUSPO$\underline{T}$: See Coordinate Transformations.

CEARTH: Value of the cosine of the minimum angle between the star and VEART$\underline{H}$ allowed in "P52C" logic (i.e. in "OCCULT"), scale factor B2, computed in "S50". It is stored in push-down list location 14D, which also has notation "CSS".

CMOON: Value of the cosine of the minimum angle between the star and VMOO$\underline{N}$ allowed in "OCCULT", scale factor B2, computed in "S50". It is stored in push-down list location 18D.

CSUN: Value of the cosine of the minimum angle between the star and VSU$\underline{N}$ allowed in "OCCULT", scale factor B2, set to K_{cssun} in "S50" (a "variable" for indexing convenience in "OCCULT"). It is stored in push-down list location 16D.

GCOM$\underline{P}$: See IMU Computations.

IMODES30: See IMU Computations.

IMUCADR: See IMU Computations.

K_{1dc}: Constant, program notation "1/C", scale factor B-6, value $4.2696E-5 \times 2^{-1}$. Value corresponds to $2^7/(9.835712E8 \times 10^{-2} \times 0.3048)$, where first term (numerator) combines the B-6 scale factor and the 2^{-1} for constant, second is velocity of light (fps), third converts to centi-seconds, and 4th converts to meters, giving the 4.2696E-5.

$K_{1dsqrt3}$: Constant, program notation "1/SQR3", scale factor B0, value 0.57735021. Value corresponds approximately to (1/1.732051), or the reciprocal of $\sqrt{3}$. Used in "PLANET" to ensure no overflow for unit.

K_{5degs}: Constant, program notation "5DEGREES", scale factor B0, units revolutions. Value is 0.013888889, corresponding to 5 degrees.

$\underline{K}_{cat}$: Table of positions of 37 stars, program notation "CATLOG", stored as the X, Y, and Z components of a unit vector, scale factor B1. The index for the table equals six times the star serial number (since each component is double precision, six cells per star are used). See table below. Because of the nature of the interpretive index order (subtractive), the tag "CATLOG" actually is one cell after the least significant half of the z component of star #1, and #1 is the star with highest memory address.

K_{cos33}: Constant, program notation "COS33", scale factor B0, value 0.8431756920. Value corresponds to cosine of 32° 31' 23.19", the same angle as that used with [NB1NB2] (see Coordinate Transformations).

K_{css5}: Constant, program notation "CSS5", scale factor B2, value 0.2490475. True value is 0.99619, corresponding to cosine 5°.

K_{css33}: Constant, program notation "CSS33", scale factor B2, value 0.197002688. True value is 0.788010752, corresponding to cosine 38° (one half the angle reflected in K_{css66}, hence the notation).

K_{css66}: Constant, program notation "CSS66", scale factor B2, value 0.060480472. True value is 0.241921888, corresponding to cos 76° (angle gate was formerly 66°, hence the notation).

$K_{css6640}$: Constant, program notation "CSS6640", scale factor B2, value -0.15602587. True value is -0.62410348, corresponding to 0.241921888 - 0.866025368, or (cosine 76° - cosine 30°). Angle gates were formerly 66° and 40°, hence the notation.

K_{cssun}: Constant, program notation "CSSUN", scale factor B2, value 0.24148. True value is 0.96592, corresponding to cosine 15°.

K_{deg1}: Constant, program notation "DEGREE1", scale factor B0, units revolutions. Octal value is 00056_8 37722_8, corresponding to about 1.033°.

K_{deg359}: Constant, program notation "DEG359", scale factor B0, units revolutions. Octal value is 37722_8 00004_8, corresponding to about 358.9893°. The single precision versions of this constant and K_{deg1} sum to one revolution.

$\underline{K}_{eclipol}$: Vector constant, program notation "ECLIPOL", scale factor B0, value (0, -0.395319722E-4, 0.911652662E-4). Values correspond approximately to 0.993674E-4 (0, -0.3978364, 0.9174565), where terms in parentheses are - sin K_{bsubo} and cos K_{bsubo} (see Coordinate Transformations), and first term is the radian equivalent of about 20.496 arc seconds (the constant of aberration).

K_{rsube}: Constant, program notation "RSUBE", scale factor B29, units meters. Value is 6378166×2^{-29}, the Fischer equatorial radius.

K_{rsubm}: Constant, program notation "RSUBM", scale factor B29, units meters. Value is 1738090×2^{-29}, the mean lunar radius.

K_{sin33}: Constant, program notation "SIN33", scale factor B0, value 0.5376381241. Value is the sine of the angle discussed for K_{cos33}.

K_{tsgtl}: Constant, program notation "TSIGHT1", scale factor B28, units centi-seconds. Value is 24000×2^{-28}, corresponding to 240 seconds.

LANDALT, LANDLONG: See Orbital and Rendezvous Navigation.

LONG: See Coordinate Transformations.

MARK2DWN+i (i = 0-6): Set of optics mark information (double precision time, CDU_y, optics shaft, CDU_z, optics trunnion, and CDU_x respectively) loaded for downlink purposes from alignment mark for the body identified by BESTJ (loaded in "SXTSM").

MARKDOWN+i (i = 0-6): See Measurement Incorporation.

MRKBUF1: See Optics Computations.

OGC: See Coordinate Transformations.

OPTION2: See Display Interface Routines.

OPTNTYPE: See Orbital and Rendezvous Navigation.

PLANVE$\underline{C}$: Temporary storage for the output of the "PLANET" routine for the first body that is marked, scale factor B1. In P51/P53 it is in reference coordinates, while in P52/P54 it is converted to IMU coordinates before being stored.

RL$\underline{S}$: See Coordinate Transformations.

SA$\underline{X}$: Unit vector, scale factor B1, giving the direction of the optics shaft axis: stars are considered visible (for selection purposes) if they lie within 38° of this axis. The quantity is expressed in the reference coordinate system.

STARA$\underline{C}$, STARA$\underline{D}$, STARB$\underline{C}$, STARB$\underline{D}$: See Coordinate Transformations.

STARCODE: Single precision cell, scale factor B14, displayed in R1 by N70/N71, and used to contain the serial number of the celestial body being sighted (0 for "planet", N88 input; $01\text{-}45_8$ for star in $\underline{K}_{cat}$; 46_8 for sun; 47_8 for earth; and 50_8 for moon). Six times this number (for double precision vector indexing reasons) is stored in BESTI or BESTJ.

STARIND: Single precision cell, scale factor B14, used to cause two stars to be processed (or other bodies). If it is 1, BESTJ is used; if 0, BESTI. Initialized to 0 in "P51B" (P51/P53) for first body and incremented to 1 for second; initialized to 1 in "R51" (P52/P54) for first body and decremented to 0 for second. Also 0 in "R00".

STARSAV1: Temporary storage for sighting information (unit vector in IMU coordinates), scale factor B1. Loaded in "P51B" with sighting vector to first body; in "R51", loaded briefly with reference-coordinate information (from "PLANET") for first body, and then loaded with sighting vector to first body. The reference-coordinate information is used in R52 for optics pointing purposes.

STARSAV2: Temporary storage for sighting information (unit vector in IMU coordinates), scale factor B1. See STARSAV1 (STARSAV2 information is for second body instead of first, of course).

STARSAV3: Cells used by N88 for retention of "planet" information. N88 treats decimal input as double precision fraction, and since a unit vector is formed before quantity used (with components previously multiplied by $K_{ldsqrt3}$), the DSKY input can be with arbitrary (but consistent) scaling.

T_{sight}: Value of sighting time, scale factor B28, units centi-seconds. It is loaded in "SXTSM" with the time at which the optics mark was made (for use in "PLANET" for computing earth, moon, or sun positions at that time in "S50" if required). It is also used in "P52B" to retain the input time for computation of the landing-site orientation, and in "P52C" to contain present time incremented by K_{tsgtl} for "S50" use in determining "OCCULT" body positions.

VEARTH: Quantity computed in "S50" as the earth position (scale factor B1, a unit vector, when exit from routine) at time T_{sight}. It has storage address the same as "STARAD" (hence at start of "PLANET", VSUN and VEARTH are traded as far as storage locations are concerned, for convenience in subsequent indexing).

VELdC: Vector, scale factor B1, giving "aberration correction vector" (see $\underline{K}_{eclipol}$) defined as "velocity of vehicle with respect to sun divided by velocity of light." It is computed in "S50" and used in "PLANET". Also computed and used in "UTAREA1".

VMOON: Quantity computed in "S50" as the moon position (scale factor B1, a unit vector, when exit from routine) at time T_{sight}. It has storage address the same as "STARAD" + 12.

VSUN: Quantity computed in "S50" as the sun position (scale factor B1, a unit vector, when exit from routine) at time T_{sight}. It has storage address the same as "STARAD" +6 (see VEARTH).

$\underline{X}_{smd}$, $\underline{Y}_{smd}$, $\underline{Z}_{smd}$: Unit vectors, scale factor B1, giving desired stable member orientation. Loaded by "P52B" or "S40.2,3(B)", or can be loaded by P27 ($\underline{X}_{smd}$ address is 0306_8, the same as "UPBUFF" +2).

Values of $\underline{K}_{cat}$ (Star Table)

Display	Index	X Component	Y Component	Z Component	Identification
01	6	0.8747608555	0.0264803244	0.4838307948	α Andromedae
02	12	0.9342466124	0.1739271769	-0.3113400137	β Ceti
03	18	0.4773424940	0.1168141178	0.8709182540	γ Cassiopeiae
04	24	0.4918322686	0.2207092653	-0.8422520048	α Eridani
05	30	0.0128995818	0.0078096205	0.9998862988	α Ursae Minoris
06	36	0.5448995598	0.5317389073	-0.6483349473	θ Eridani
07	42	0.7028937840	0.7078988678	0.0694227718	α Ceti
10	48	0.4101921571	0.4989947555	0.7633784305	α Persei
11	54	0.3502769546	0.8927907521	0.2832502918	α Tauri
12	60	0.2007345455	0.9691236271	-0.1431958012	β Orionis
13	66	0.1367280274	0.6814364033	0.7189922632	α Aurigae
14	72	-0.0616090080	0.6031289258	-0.7952608559	α Carinae
15	78	-0.1824340636	0.9404062129	-0.2869738089	α Canis Majoris
16	84	-0.4122762536	0.9063680102	0.0923326629	α Canis Minoris
17	90	-0.3613649782	0.5745656444	-0.7343634472	γ' Velorum
20	96	-0.4661511162	0.4772744503	0.7449243154	ι Ursae Majoris
21	102	-0.7745052221	0.6149043688	-0.1484394758	α Hydrae
22	108	-0.8610673209	0.4632386329	0.2096974909	α Leonis
23	114	-0.9657334280	0.0521664164	0.2542392790	β Leonis
24	120	-0.9524366380	-0.0597677121	-0.2988181236	γ Corvi
25	126	-0.4521486548	-0.0495728431	-0.8905639377	α' Crucis
26	132	-0.9168160791	-0.3506241694	-0.1910784362	α Virginis
27	138	-0.5812217481	-0.2911759648	0.7598669864	η Ursae Majoris
30	144	-0.6895375091	-0.4185354938	-0.5910719617	θ Centauri
31	150	-0.7860186221	-0.5221457573	0.3309660611	α Bootis
32	156	-0.5324545035	-0.7163035719	0.4510004372	α Coronae Borealis
33	162	-0.3511952476	-0.8242322268	-0.4441881743	α Scorpii
34	168	-0.1142900725	-0.3400201762	-0.9334474056	α Trianguli Australis
35	174	-0.1120382967	-0.9695442116	0.2177876068	α Ophiuchi
36	180	0.1219537054	-0.7702168243	0.6260138474	α Lyrae
37	186	0.2074286490	-0.8718956797	-0.4435890882	σ Sagittarii
40	192	0.4540867784	-0.8777450759	0.1528685033	α Aquilae
41	198	0.5524232365	-0.7930716636	-0.2566435348	β Capricorni
42	204	0.3205423120	-0.4434583652	-0.8370169081	α Pavonis
43	210	0.4542117996	-0.5390337930	0.7093195408	α Cygni
44	216	0.8141988673	-0.5553601830	0.1692786800	ε Pegasi
45	222	0.8345006310	-0.2388718657	-0.4965369357	α Piscis Austrini

The "Display" column gives the star number that is displayed by the program (as an octal quantity). The "Index" column gives the value of BESTI (or BESTJ) for the corresponding star, and is equal to six times the star serial number (i.e. six times the decimal equivalent of the first column).

A display of 00 indicates an N88 input; displays of 46, 47, and 50 are for sun, earth, and moon respectively.

Mathematical Functions

Sine

Routine entered with argument in $MPAC_{dp}$, scale factor B0, units revolutions. Leaves $MPAC_{dp}$ with sine of argument, scale factor B1.

$X = MPAC_{dp}$

Perform "SICOM"

$MPAC_{dp} = TS$

Return

SICOM

If $|X| \geq \frac{1}{2}$, $X = \frac{1}{2}\,\mathrm{sgn}\,X - X$

If $|X| \geq \frac{1}{4}$, $X = \frac{1}{2}\,\mathrm{sgn}\,X - X$

$x = X$, rescaled to scale factor B-1 revolutions, i.e. B1 in $(\pi/2)$ units.

$TS = K_{sn1}\,x + K_{sn3}\,x^3 + K_{sn5}\,x^5 + K_{sn7}\,x^7 + K_{sn9}\,x^9$

Return

Constant	$\pi/2$ units Scaling	Stored Value	True Value x $(2/\pi)^i$
K_{sn1}	B2	0.39269 90796	0.99999 9995
K_{sn3}	B0	-0.64596 37111	-0.16666 6567
K_{sn5}	B-2	0.31875 8717	0.00833 3025
K_{sn7}	B-4	-0.07478 0249	-0.00019 8074
K_{sn9}	B-6	0.00969 4988	0.00000 2603

According to the program comments, the constants are derived from a Hastings series.

Cosine

Routine entered with argument in $MPAC_{dp}$, scale factor B0, units revolutions. Leaves $MPAC_{dp}$ with cosine of argument, scale factor B1.

$X = \frac{1}{4} - |MPAC_{dp}|$

Perform "SICOM"

$MPAC_{dp} = TS$

Return

Single Precision Sine ($\sin_{sp}$)

Routine entered with argument in computer hardware accumulator, ACC, single precision, scale factor B-1, units revolutions. Leaves sine in hardware accumulator, scale factor B0.

$X = ACC$

Perform "SPSICOM"

$ACC = TS$

Return

SPSICOM

If $|X| \geq \frac{1}{4}$, $X = \frac{1}{2} \text{ sgn } X - X$

If $|X| \geq \frac{1}{4}$:

$TS = MAX \text{ sgn } X$

Return

$x = X$, rescaled to scale factor B-2 revolutions, i.e. B0 in $(\pi/2)$ units.

$TS = K_{c1sp}\, x + K_{c3sp}\, x^3 + K_{c5sp}\, x^5$, limited in magnitude < 1

Return

Constant	Nominal Value	Stored Value	True Value x $(2/\pi)^i$
K_{c1sp}	0.7853134	0.78533935	0.999925
K_{c3sp}	-0.3216147	-0.32159423	-0.165951
K_{c5sp}	0.0363551	0.03637695	0.007608

All constants have scale factor B1, for $\pi/2$ units.

Single Precision Cosine ($\cos_{sp}$)

Routine entered with argument in computer hardware accumulator, ACC, single precision, scale factor B-1, units revolutions. Leaves cosine in hardware accumulator, scale factor B0.

$X = ACC + \frac{1}{4}$

If $|X| \geq \frac{1}{2}$:

$X = -(X - \frac{1}{2} \text{ sgn } X)$

Perform "SPSICOM"

$ACC = TS$

Return

Arc Sine

Routine entered with argument in $MPAC_{tp}$, scale factor B1. Leaves $MPAC_{dp}$ with arc sine of argument, scale factor B0, units revolutions, in range $\pm \frac{1}{4}$ (i.e. $\pm 90^o$).

$X = MPAC_{tp}$

Perform "ARCCOM"

$MPAC_{dp} = \frac{1}{4} - TS$

Return

ARCCOM

If $|X| = 0$, $TS = \frac{1}{4}$; Return

$TS_1 = X$

$X = |X|$

If $X \geq (1 + 2^{-12})$, or 1.000244, or if least significant half of X is negative and $X > (1 + 2^{-13})$, or 1.000122:

Perform "ALARM" (pattern 1301_8)

If $X > 1$, set $TS = 0$ (for TS_1 positive) or $TS = \frac{1}{2}$ (for TS_1 negative); Return

$TS = (1 - X)^{\frac{1}{2}}$ (scale factor B1 when enter square root routine)

$$TS = TS(K_{as0} + K_{as1} X + K_{as2} X^2 + K_{as3} X^3 + K_{as4} X^4 + K_{as5} X^5 + K_{as6} X^6 + K_{as7} X^7)$$

If TS_1 negative, $TS = \frac{1}{2} - TS$

Return

Constant	True Value	$\sqrt{2}\,\pi$ x True Value
K_{as0}	0.35355 3385	1.57079 6302
K_{as1}	-0.04830 17006	-0.21459 8801
K_{as2}	0.02002 73085	0.08897 8987
K_{as3}	-0.01129 31863	-0.05017 4305
K_{as4}	0.00695 311612	0.03089 1881
K_{as5}	-0.00384 617957	-0.01708 8126
K_{as6}	0.00150 1297736	0.00667 0090
K_{as7}	-0.00028 4160334	-0.00126 2491

All constants are stored with scale factor B-I in program (e.g. K_{as5} has scale factor B-5); the $\sqrt{2}$ factor is required because (1 - X) is scaled at B1 when take its square root. The numbers in the last column agree closely with published Hastings series values.

Arc Cosine

Routine entered with argument in $MPAC_{tp}$, scale factor B1. Leaves $MPAC_{dp}$ with arc cosine of argument, scale factor B0, units revolutions, in range 0 - $\frac{1}{2}$ (i.e. 0° - 180°).

$X = MPAC_{tp}$

Perform "ARCCOM"

$MPAC_{dp} = TS$

Return

Square Root

Routine entered with argument in $MPAC_{tp}$, and with scale factor (in general) an even number. Leaves $MPAC_{tp}$ with square root of argument, having scale factor one-half the scale factor of argument (a $\sqrt{2}$ factor must be accounted for elsewhere if the scale factor was odd when enter the routine).

$X = MPAC_{tp}$, considered to have scale factor B0

If $X = 0$, $MPAC_{tp} = 0$; Return

If $X < -2^{-14}$, proceed to "POODOO" (pattern 21302_8)

If $X < 0$, $MPAC_{dp} = 0$; Return

TS = X, normalized to lie between $\frac{1}{4}$ and < 1. This is accomplished by shifting X left in increments of 2 shifts, and storing the required number of shifts in TS_1 (an even number ≥ 0)

If $TS \geq \frac{1}{2}$, $TS_2 = 0.5884\ TS_{sp} + 0.4192$

If $TS < \frac{1}{2}$, $TS_2 = 0.8324\ TS_{sp} + 0.2974$

$TS_2 = (\frac{1}{2}\ TS_{sp})\ /\ TS_2 + \frac{1}{2}\ TS_2$ (TS_2 single precision)

$TS = \frac{1}{2}\ TS_2 + \frac{1}{2}\ TS\ /\ TS_2$

If $TS \geq 1$, $TS = (1 - 2^{-28})$

$MPAC_{tp}$ = TS, shifted right by ($\frac{1}{2}\ TS_1$) places

Return

Natural Logarithm ($\log_e$)

This routine is entered from "RANGER", with the argument in $MPAC_{dp}$, scale factor B0. It leaves the natural logarithm of the reciprocal of the argument (or the complement of the logarithm of the argument) in $MPAC_{dp}$, scale factor B5.

LOG

TS = $MPAC_{dp}$, normalized to lie between $\frac{1}{2}$ and < 1. This is accomplished by shifting $MPAC_{dp}$ left and storing the required number of shifts in TS_1 (a number ≥ 0).

$X = 1 - TS$ (The "1" actually is $1 - 2^{-28}$)

$MPAC_{dp} = K_{cog1}\ TS_1 + K_{og1}\ X + K_{og2}\ X^2 + K_{og3}\ X^3$

Return (with value of $-\log_e$ of original $MPAC_{dp}$)

All constants are stored with scale factor B5. The value of K_{cog1} corresponds to $\log_e 2$.

Constant	Stored Value	True Value
K_{cog1}	0.0216608494	0.69314718056
K_{og1}	0.031335467	1.00273494
K_{og2}	0.0130145859	0.4164667488
K_{og3}	0.0215738898	0.6903644736

Measurement Incorporation

INCORP1

$\underline{Z}_0 = [W_0]\ \mathrm{BVECTOR}\underline{}_0 + [W_3]\ \mathrm{BVECTOR}\underline{}_1 + [W_6]\ \mathrm{BVECTOR}\underline{}_2$

$\underline{Z}_1 = [W_1]\ \mathrm{BVECTOR}\underline{}_0 + [W_4]\ \mathrm{BVECTOR}\underline{}_1 + [W_7]\ \mathrm{BVECTOR}\underline{}_2$

$\underline{Z}_2 = [W_2]\ \mathrm{BVECTOR}\underline{}_0 + [W_5]\ \mathrm{BVECTOR}\underline{}_1 + [W_8]\ \mathrm{BVECTOR}\underline{}_2$

If bit 9(DMENFLG) of FLAGWRD5 = 0: (means 6-dimensional incorporation)

$\underline{Z}_2 = 0$

$\mathrm{TRIPA} = |\underline{Z}_0|^2 + |\underline{Z}_1|^2 + |\underline{Z}_2|^2 + \mathrm{VARIANCE}$

$\mathrm{TS} = \sqrt{\mathrm{TRIPA}\ \ \mathrm{VARIANCE}}$ computed quasi-floating point (TRIPA rounded to double precision before multiplication use)

$\mathrm{GAMMA} = 1/(\mathrm{TS} + \mathrm{TRIPA})$ computed quasi-floating point

$\mathrm{DELQDA} = \mathrm{DELTAQ} / \mathrm{TRIPA}$ computed quasi-floating point

Set TS_1 to the largest $|\underline{Z}_i|$ $(i = 0, 1, 2)$ (tag here "NEWZCOMP")

TS_2 = (number of leading zeros in TS_1) - 2 (value used for subsequent scaling)

Shift each $\underline{Z}_i$ left by TS_2 places $(i = 0, 1, 2)$

$\mathrm{OMEGA}\underline{}_0 = \underline{Z}_0\ [W_0] + \underline{Z}_1\ [W_1] + \underline{Z}_2\ [W_2]$

$\mathrm{OMEGA}\underline{}_1 = \underline{Z}_0\ [W_3] + \underline{Z}_1\ [W_4] + \underline{Z}_2\ [W_5]$

$\mathrm{OMEGA}\underline{}_2 = \underline{Z}_0\ [W_6] + \underline{Z}_1\ [W_7] + \underline{Z}_2\ [W_8]$

If bit 9(DMENFLG) of FLAGWRD5 = 0:

$\mathrm{OMEGA}\underline{}_2 = 0$

$\mathrm{DELTAX}\underline{}_0 = \mathrm{DELQDA}\ \ \mathrm{OMEGA}\underline{}_0$ computed

$\mathrm{DELTAX}\underline{}_1 = \mathrm{DELQDA}\ \ \mathrm{OMEGA}\underline{}_1$ quasi-floating

$\mathrm{DELTAX}\underline{}_2 = \mathrm{DELQDA}\ \ \mathrm{OMEGA}\underline{}_2$ point

Return

INCORP2

EGRESS = Return address

Perform "INTSTALL"

$\mathrm{OMEGAM}\underline{}_0 = \mathrm{GAMMA}\ \ \mathrm{OMEGA}\underline{}_0$

$\mathrm{OMEGAM}\underline{}_1 = \mathrm{GAMMA}\ \ \mathrm{OMEGA}\underline{}_1$

$\mathrm{OMEGAM}\underline{}_2 = \mathrm{GAMMA}\ \ \mathrm{OMEGA}\underline{}_2$

Set bit 13(INTGRAB) of FLGWRD10 = 1 (means restart phase change)

$[W_0] = [W_0] - (\underline{Z}_0)\ \text{OMEGA}\underline{\text{M}}_0$

$[W_1] = [W_1] - (\underline{Z}_1)\ \text{OMEGA}\underline{\text{M}}_0$

$[W_3] = [W_3] - (\underline{Z}_0)\ \text{OMEGA}\underline{\text{M}}_1$

$[W_4] = [W_4] - (\underline{Z}_1)\ \text{OMEGA}\underline{\text{M}}_1$

NOTE: Notation means computation carried out as:

$$[W_0] = [W_0] - \begin{bmatrix} Z_{0,x} \\ Z_{0,y} \\ Z_{0,z} \end{bmatrix} \text{OMEGA}\underline{\text{M}}_0$$

If bit 9(DMENFLG) of FLAGWRD5 = 1: (9-dimensional incorporation)

$[W_2] = [W_2] - (\underline{Z}_2)\ \text{OMEGA}\underline{\text{M}}_0$

$[W_5] = [W_5] - (\underline{Z}_2)\ \text{OMEGA}\underline{\text{M}}_1$

$[W_6] = [W_6] - (\underline{Z}_0)\ \text{OMEGA}\underline{\text{M}}_2$

$[W_7] = [W_7] - (\underline{Z}_1)\ \text{OMEGA}\underline{\text{M}}_2$

$[W_8] = [W_8] - (\underline{Z}_2)\ \text{OMEGA}\underline{\text{M}}_2$

All W-matrix updating computations performed quasi-floating point

$\text{TX78}\underline{9} = \text{X78}\underline{9} + \text{DELTA}\underline{\text{X}}_2$ (Tag here "FAZC")

If bit 8(CSMUPDAT) of FLAGWRD1 = 1:

Perform "MOVEPCSM"

If bit 8(CSMUPDAT) of FLAGWRD1 = 0:

Perform "MOVEPLEM"

Reset overflow indicator

X2 = 0

If bit 12(CMOONFLG) of FLAGWRD8 = 1:

X2 = 2

$\text{T}\underline{\text{S}} = \text{TDELTA}\underline{\text{V}} + (\text{DELTA}\underline{\text{X}}_0$ shifted left 7 + X2 places)

If overflow has taken place since indicator reset:

$\text{RC}\underline{\text{V}} = \text{RC}\underline{\text{V}} + \text{DELTA}\underline{\text{X}}_0$

$\text{VC}\underline{\text{V}} = \text{VC}\underline{\text{V}} + \text{DELTA}\underline{\text{X}}_1$

PBODY = X2 (0 for earth, 2 for moon)

(If overflow has taken place since indicator reset):

Perform "RECTIFY"

Proceed to "FAZAB3"

$\text{TDELTA}\underline{\text{V}} = \text{T}\underline{\text{S}}$

$\text{T}\underline{\text{S}} = \text{TNU}\underline{\text{V}}$ + ($\text{DELTA}\underline{\text{X}}_1$ shifted left 4 + X2 places)

If overflow has taken place:

$\text{VC}\underline{\text{V}} = \text{VC}\underline{\text{V}} + \text{DELTA}\underline{\text{X}}_1$

PBODY = X2

Perform "RECTIFY"

Proceed to "FAZAB3"

$\text{TNU}\underline{\text{V}} = \text{T}\underline{\text{S}}$

Proceed to "FAZAB3"

<u>FAZAB3</u>

If bit 8(CSMUPDAT) of FLAGWRD1 = 1:

Perform "MOVEACSM"

If bit 1(AVEMIDSW) of FLAGWRD9 = 0:

$\underline{\text{R}} = \text{RC}\underline{\text{V}} + \text{TDELTA}\underline{\text{V}}$ (X2 used to determine

$\underline{\text{V}} = \text{VC}\underline{\text{V}} + \text{TNU}\underline{\text{V}}$ necessary shifts)

$T_{pptm} = T_{et}$

If bit 8(CSMUPDAT) of FLAGWRD1 = 0:

Perform "MOVEALEM"

$\underline{\text{R}}_{other} = \text{RC}\underline{\text{V}} + \text{TDELTA}\underline{\text{V}}$ (X2 used to determine

$\underline{\text{V}}_{other} = \text{VC}\underline{\text{V}} + \text{TNU}\underline{\text{V}}$ necessary shifts)

If bit 9(DMENFLG) of FLAGWRD5 = 1: (9-dimensional incorporation)

$\text{X78}\underline{\text{9}} = \text{TX78}\underline{\text{9}}$

QPRET = EGRESS

Proceed to "INTWAKE" (returns to address in QPRET)

V67CALL Established by "V67" for a verb 67.

Reset overflow indicator

Perform "INTSTALL"

WWOPT = 0

$$WWPOS = \sqrt{|\underline{W}_0|^2 + |\underline{W}_3|^2 + |\underline{W}_6|^2 + |\underline{W}_9|^2 + |\underline{W}_{12}|^2 + |\underline{W}_{15}|^2}$$

(i.e. $\underline{W}_{3i}$, $i = 0\text{-}5$)

$$WWVEL = \sqrt{|\underline{W}_{27}|^2 + |\underline{W}_{30}|^2 + |\underline{W}_{33}|^2 + |\underline{W}_{36}|^2 + |\underline{W}_{39}|^2 + |\underline{W}_{42}|^2}$$

(i.e. $\underline{W}_{3i}$, $i = 9\text{-}14$)

If overflow has taken place since indicator reset:

WWPOS = +MAX ($2^{19} - 2^{-9}$ meters)

WWVEL = +MAX (N99 R2 display 328.1 fps)

If WWPOS - $K_{ft999} \geq 0$:

WWPOS = K_{ft999} (constant gives display of 99997 feet)

Perform "INTWAKE" (returns to next line)

TS = 0699_{vn}

Proceed to "GOXDSPF": if terminate, proceed to "ENDEXT"
if proceed, proceed
otherwise, proceed to previous line

If WWOPT = 0: (no change in W-matrix initialization cells desired)

Proceed to "ENDEXT"

TS_1 = $WWPOS_{sp}$

TS_2 = $WWVEL_{sp}$

If WWOPT = 2:

WORBPOS = TS_1

WORBVEL = TS_2

If WWOPT > 2:

$WMIDPOS = TS_1$

$WMIDVEL = TS_2$

If WWOPT < 2:

$WRENDPOS = TS_1$

$WRENDVEL = TS_2$

Set bit 6(ORBWFLAG) of FLAGWRD3 = 0 (must set RENDWFLG = 0 by e.g. V93E)

Proceed to "ENDEXT"

<u>GETUM</u>

Perform "SXTSM" starting at third line (STARIND = 0 from "R00")

$\underline{TS} = \underline{TS}\ [REFSMMAT]$

Return

<u>BVECTORS</u>

$BVECTO\underline{R}_0 = unit(USTA\underline{R} * unitRCL\underline{P})$

$USTA\underline{R} = BVECTO\underline{R}_0$

$DELTAQ = K_{2pi} \left| RCL\underline{P} \right| \left(\cos^{-1} (BVECTO\underline{R}_0 \cdot U\underline{M}) - \frac{1}{4} \right)$ $(\frac{1}{4} = 90^{\circ})$

$BVECTO\underline{R}_1 = 0$

$BVECTO\underline{R}_2 = 0$

Return

<u>9DWTO6DW</u> (Entered from "S22.981X" and "S22GTP")

9DWXX = Return address

9DWP = 0

9DWJ = 29

9DWI = 9DWJ

Proceed to "9DWEPCAL"

<u>9DWEPCAL</u>

Perform "ROWDOT"

$EMATRIX_{20-9DWP}$ = TS

9DWP = 9DWP + 1

If 9DWI $\neq$ 0:

  9DWI = 9DWI - 1

  If 9DWI = 26:

    9DWI = 2

  Proceed to "9DWEPCAL"

If 9DWJ $\neq$ 0:

  9DWJ = 9DWJ - 1

  If 9DWJ = 26:

    9DWJ = 2

  9DWI = 9DWJ

  Proceed to "9DWEPCAL"

9DWJ = 29

9DWP = 0

9DWI = 0

9DWN = 0

$W_i = 0$ $(i = 0 - 53)$

Proceed to "9DWI=JA"

<u>9DWI=JA</u>

9DWI = 9DWJ

Perform "ROWDOT"

$TS = EMATRIX_{20-9DWP} - TS$

$9DWP = 9DWP + 1$

$X1 = -(9DWI + 9DWN)$

If $TS \geq 0$:

$TS_1 = \sqrt{TS}$

If $TS < 0$:

$TS_1 = 0$

$W_{-X1} = TS_1$

$WORKW = TS_1$

If $9DWJ = 0$:

$W_i = 0 \quad (i = 54 - 80)$

Proceed to address specified by 9DWXX

Proceed to "TST2I=0"

<u>TST2I=0</u>

If $9DWI = 0$:

$9DWN = 9DWN + 3$

$9DWJ = 9DWJ - 1$

If $9DWJ = 26$:

$9DWJ = 2$

Proceed to "9DWI=JA"

$9DWI = 9DWI - 1$

If $9DWI = 26$:

$9DWI = 2$

Perform "ROWDOT"

$TS = (EMATRIX_{20-9DWP} - TS) / WORKW$

$9DWP = 9DWP + 1$

$X1 = -(9DWI + 9DWN)$

If overflow has taken place since indicator reset:

$TS = 0$

$W_{-X1} = TS$

Proceed to "TST2I=0"

ROWDOT

Reset overflow indicator

$TS_1 = 8$

$X1 = -9DWI$

$X2 = -9DWJ$

$TS = 0$

Proceed to "ROWDOT1"

ROWDOT1

$TS = W_{-X1} \; W_{-X2} + TS$ (modulo computer word length)

If no overflow has taken place:

$X1 = X1 - 3$ (scaled B13)

$X2 = X2 - 3$ (scaled B13)

If $TS_1 > 0$:

$TS_1 = TS_1 - 1$

Proceed to "ROWDOT1"

Return (TS set to sum)

Set bit 6(ORBWFLAG) of FLAGWRD3 = 0 (overflow indicator reset)

Return

R22 (Established by "NDUTINPT" for P20 option 0/4, and entered due to restart group 2.7)

Change priority of present job to 26_8 (restart group 2.7 has 10_8 for establishing "R22")

MRKBUF1 = -3

VHFTIME = T_{now}

Proceed to "REND1"

REND1

Change restart group 2 to cause a start at next line (subsequent restart logic not shown unless significant for normal logic)

Perform "WAITONE"

If MRKBUF1 $\leq$ 0: (Tag here "REND1A")

Proceed to "REND3"

Inhibit interrupts

MARKTIME+i = MRKBUF1+i (i = 0 - 6)

MRKBUF1 = -3

Set bit 7(R22CAFLG) of FLAGWRD9 = 1 (for "MKREJECT" use)

Release interrupts

Set bit 8(VHFSOURC) of FLAGWRD9 = 0

Proceed to "REND4"

REND3

Change restart group 2 to cause a start at next line

Perform "WAITONE"

If bit 9(VHFRFLAG) of FLAGWRD9 = 0:

Proceed to 3rd line of "REND1"

If bit 9(RANGFLAG) of FLGWRD10 = 0: (set 0, then 1 if appropriate, in "CRS61.1", run at lower priority than "R22" (hence R22 could interrupt between 0 and 1 setting))

Proceed to 3rd line of "REND1"

If (T_{now} - $K_{60secdp}$ - VHFTIME) < 0:

Proceed to 3rd line of "REND1"

Inhibit interrupts (Tag here "RANGERD")

Set bits 4-1 (Radar selection) of channel 13 = 0

Proceed to "RADSTART"

RADSTART

TS = channel 4

TS_1 = (bits 5-1 of TS)

RADDEL = bits 5-1 of (00040_8 - TS_1)

If RADDEL $\leq 00002_8$: (bit 6 of channel 4, in phase with 100 pps signal used for radar control, about to be incremented)

Proceed to "RADSTART"

Set bits 1 and 4 (VHF select and Radar initiate) of channel 13 = 1

RADTIME = - TS

Release interrupts

MARKTIME = T_{now}

Set bit 10 (SKIPVHF) of FLAGWRD2 = 0 (set 1 in "STARTSB2")

End of job

VHFREAD (Entered based on program interrupt #9, Radar Rupt)

If bit 10(SKIPVHF) of FLAGWRD2 = 1: (set in "STARTSB2")

Resume

If bit 7(UPDATFLG) of FLAGWRD1 = 1:

If bit 12(PDSPFLAG) of FLAGWRD4 = 0: (R61 not doing R60)

RM = RNRAD

If bit 2(VHF Data Good complement) of channel 33 = 0:

Establish "RANGERD1" (priority 26_8)

Resume

$VHFRANGE_{sp} = 2_8$ (bit 2 of channel 33 = 1, flag "LIGHTON" to set Tracker light)

Skip next line

$VHFRANGE_{sp} = 0$ (UPDATFLG = 0 or PDSPFLAG = 1) (Tag "BYPASS1")

Establish "LIGHTON" (priority 26_8)

Resume

LIGHTON (Established by "VHFREAD" for unsuccessful radar read)

If $VHFRANGE_{sp} \neq 0$:

Set bit 8(Tracker) of DSPTAB+11 = 1, and flag for output at next opportunity

VHFTIME = MARKTIME

Proceed to "REND1"

RANGERD1 (Established by "VHFREAD" for successful radar read)

Inhibit interrupts

Set bit 8(Tracker) of DSPTAB+11 = 0, and flag for output at next opportunity

Set bit 7(OCDUFBIT) of OPTMODES = 1

Release interrupts

If RM = +0:

Proceed to 3rd line of "LIGHTON"

If bit 15 of RM = 0:

$TS = K_{convrnge} (RM, 0)$

If bit 15 of RM = 1: (including case of 77777_8)

TS = bits 14-1 of RM

$TS = K_{convrnge} (TS, 0) + 2^{14} K_{convrnge}$

VHFRANGE = TS (Tag here "RANGERD2")

Set bit 8(VHFSOURC) of FLAGWRD9 = 1

VHFTIME = MARKTIME

Proceed to "REND4"

REND4 (Entered from "RANGERD1" for VHF measurement, and from "REND1" for an optics measurement)

Perform "SETINTG"

If bit 8(CSMUPDAT) of FLAGWRD1 = 1:

Set bit 3(CSMINTSW) of FLAGWRD3 = 0

Perform "INTEGRV"

Perform "SETINTG"

If bit 1(RENDWFLG) of FLAGWRD5 = 1:

Set bit 1(WMATINT) of FLAGWRD3 = 1

If bit 8(CSMUPDAT) of FLAGWRD1 = 0:

Set bit 3(CSMINTSW) of FLAGWRD3 = 0

Perform "INTEGRV" (when exit, LM and CSM vectors both integrated "permanently" to MARKTIME)

X2 = 0

If bit 12(CMOONFLG) of FLAGWRD8 = 1:

X2 = -2

Set bit 6(ORBWFLAG) of FLAGWRD3 = 0

CSMPOS = $RC\underline{V}_{cm}$ + (DELTA$\underline{V}_{cm}$ shifted right 7 - X2 places)

Set bit 11(FSTINCRP) of FLAGWRD5 = 1

LEMPOS = $RC\underline{V}_{lm}$ + (DELTA$\underline{V}_{lm}$ shifted right 7 - X2 places)

If bit 7(AUTOSEQ) of FLGWRD10 = 1:

Proceed to "AUTOW" (exits to "REND5C" if initialize, or "REND7" if don't)

If bit 1(RENDWFLG) of FLAGWRD5 = 1:

Set bit 1(RENDWFLG) of FLAGWRD5 = 1 (redundant, Tag "REND6")

Proceed to "REND7"

Proceed to "REND5C"

REND5C

VHFCNT = 0

TRKMKCNT = 0 (as "least significant half" of VHFCNT)

$W_i = 0$ (i = 0 - 17) (i.e. $[W_0]$ and $[W_1]$)

$W_i = 0$ (i = 27 - 44) (i.e. $[W_3]$ and $[W_4]$)

W_i = WRENDPOS (i = 0, 4, 8, diagonal elements of $[W_0]$)

W_i = WRENDVEL (i = 36, 40, 44, diagonal elements of $[W_4]$)

Set bit 8(P35FLAG) of FLGWRD10 = 0

AGEOFW = MARKTIME

Set bit 4(PTV93FLG) of FLGWRD10 = 0

COUNT3MK = 1

Set bit 1(RENDWFLG) of FLAGWRD5 = 1

Proceed to "REND7"

REND7

RCL$\underline{P}$ = LEMPO$\underline{S}$ - CSMPO$\underline{S}$

If bit 8(VHFSOURC) of FLAGWRD9 = 0: (i.e. have optics data)

UC$\underline{L}$ = unitRCL$\underline{P}$

If bit 11(FSTINCRP) of FLAGWRD5 = 1:

X1 = - "MARKTIME"

Perform "GETUM"

U$\underline{M}$ = T$\underline{S}$

T$\underline{S}$ = UC$\underline{L}$ * U$\underline{M}$

If all components of T$\underline{S}$ $< 2^{-19}$ rad: (so unit vector overflow)

Proceed to "RENDISP3"

USTA$\underline{R}$ = unitT$\underline{S}$

Perform "BVECTORS" (Tag here "REND9")

If bit 8(CSMUPDAT) of FLAGWRD1 = 0:

$BVECTOR_0$ = - $BVECTOR_0$

If bit 14(R21MARK) of FLAGWRD2 = 1: (i.e. prime optics)

$TS_1 = K_{sxtvar} + K_{imuvar}$

If bit 14(R21MARK) of FLAGWRD2 = 0: (i.e. R23 optics)

$TS_1 = C_{altvar} + K_{imuvar}$ (Tag here "REND15")

If bit 8(VHFSOURC) of FLAGWRD9 = 1: (i.e. have VHF ranging data)

$\underline{BVECTOR}_0$ = unit$\underline{RCLP}$ (Tag here "REND14")

$\underline{BVECTOR}_1 = 0$

$\underline{BVECTOR}_2 = 0$

TS = VHFRANGE (scale factor B27)

If bit 12(CMOONFLG) of FLAGWRD8 = 0:

Shift TS right 2 places (rescale to B29)

Set bit 11(FSTINCRP) of FLAGWRD5 = 1

DELTAQ = TS $- |\underline{RCLP}|$

If bit 8(CSMUPDAT) of FLAGWRD1 = 1:

$\underline{BVECTOR}_0 = - \underline{BVECTOR}_0$

$TS_1 = C_{rvar}$

$TS = (|\underline{RCLP}|^2)\ TS_1$ Computed quasi-floating point (Tag here "REND10")

X2 = 0

If bit 12(CMOONFLG) of FLAGWRD8 = 1:

X2 = -2

If X2 = 0:

Shift TS left 2 places (to scale factor B40)

If X2 = -2:

Shift TS right 2 places (to scale factor B40)

$VARIANCE = TS_{tp} + C_{intvar}$

If bit 8(VHFSOURC) of FLAGWRD9 = 1:

If $(VARIANCE + C_{rvarmin}) < 0$: ($C_{rvarmin}$ negative)

$VARIANCE = |C_{rvarmin}|$

Set bit 9(DMENFLG) of FLAGWRD5 = 0 (means 6-dimensional incorporation)

Perform "INCORP1"

If bit 11(FSTINCRP) of FLAGWRD5 = 1:

X2 = 0

If bit 12(CMOONFLG) of FLAGWRD8 = 1:

X2 = -2

N49DISP+2 = ($|DELTAX_{\underline{1}}|$ shifted right -X2 places, to B7)

N49DISP = ($|DELTAX_{\underline{0}}|$ shifted right -X2 places, to B29)

N49DISP+4 = 2

If bit 8(VHFSOURC) of FLAGWRD9 = 0: (processing optics)

If bit 12(REJCTFLG) of FLGWRD10 = 1: (set in "MKREJECT")

Proceed to "RENDISP3"

N49DISP+4 = 1

If (C_{rmax} - N49DISP) $<$ 0: ("N49DISP" same as "$BVECTOR_{\underline{0}}$")

Proceed to "RENDISP"

If (C_{vmax} - N49DISP+2) $<$ 0:

Proceed to "RENDISP"

Proceed to "REND12"

<u>REND12</u>

Perform "INCORP2" (incorporates update, same time tag as from "REND4" integration)

If bit 11(FSTINCRP) of FLAGWRD5 = 1:

OLDMKTME = MARKTIME

Set bit 5(MANEUFLG) of FLGWRD10 = 0

TS = COUNT3MK

If TS $\neq$ 0:

If (TS - 3) $\geq$ 0:

COUNT3MK = 0

If (TS - 3) $<$ 0:

COUNT3MK = COUNT3MK + 1

If bit 8(VHFSOURC) of FLAGWRD9 = 1: (Tag here "REND12A")

VHFCNT = VHFCNT + 1

Proceed to "RENDISP3"

If bit 11(FSTINCRP) of FLAGWRD5 = 0: (i.e. 2nd optics incorporation has been completed)

TRKMKCNT = TRKMKCNT + 1

Proceed to "RENDISP3"

X2 = 0

If bit 12(CMOONFLG) of FLAGWRD8 = 1:

X2 = -2

If bit 8(CSMUPDAT) of FLAGWRD1 = 1:

CSMPOS = $RC\underline{V}_{cm}$ + (DELTA$\underline{V}_{cm}$ shifted right 7 - X2 places)

If bit 8(CSMUPDAT) of FLAGWRD1 = 0:

LEMPOS = $RC\underline{V}_{lm}$ + (DELTA$\underline{V}_{lm}$ shifted right 7 - X2 places)

Set bit 11(FSTINCRP) of FLAGWRD5 = 0

Proceed to "REND7"

RENDISP

TEMPOR1 = +0

Establish "RENDISP2" (priority 27_8)

Allow performance of higher priority jobs (e.g. "RENDISP2")

If TEMPOR1 = 0: (i.e. display not answered yet)

Proceed to second previous line

If TEMPOR1 < 0: (i.e. a PRO display response)

Proceed to "REND12"

Proceed to "RENDISP3"

RENDISP2 (Established by "RENDISP")

TS = 0649_{vn}

Proceed to "PRIODSP": if terminate, proceed to "TRACKTRM"
if proceed, TS = -1 and proceed
otherwise, TS = QREG and proceed

TEMPOR1 = TS

End of job

RENDISP3 (Entered at end of each mark-processing pass)

Set bit 7(R22CAFLG) of FLAGWRD9 = 0

Set bit 12(REJCTFLG) of FLGWRD10 = 0

If bit 8(VHFSOURC) of FLAGWRD9 = 1:

Proceed to "REND1"

Proceed to "REND3"

WAITONE

POINTEX = Return address

Delay 4 seconds (by putting job to sleep via "DELAYJOB")

If bit 7(RNDVZFLG) of FLAGWRD0 = 0, End of job

If bit 13(REFSMFLG) of FLAGWRD3 = 0, End of job

If R61CNTR < 0: (R61 maneuver in process, required DSKY for R60)

Proceed to second line of "WAITONE"

If bit 7(UPDATFLG) of FLAGWRD1 = 1:

Proceed to address specified by POINTEX

If bit 5(TRACKFLG) of FLAGWRD1 = 0, End of job

Proceed to "REDOR22"

REDOR22 (Entered from "WAITONE" and due to restart group 2.13)

Set restart group 2 to phase 13 (2.13, causing "REDOR22" to be established with priority 10_8 if a restart)

Change priority of present job to 26_8

Proceed to second line of "WAITONE"

SETINTG

Perform "INTSTALL"

Set bit 5(STATEFLG) of FLAGWRD3 = 1

T_{decl} = MARKTIME

Set bits 4(CONICINT), 2(9DIMWMAT), and 1(WMATINT) of FLAGWRD3 = 0

Set bit 3(CSMINTSW) of FLAGWRD3 = 1

Return

C13STALL

If bit 4(Radar initiate) of channel 13 = 0:

Return (no VHF read in progress)

TS_3 = channel 4

$TS_3 = TS_3$ + RADTIME + 2^{14}, modulo 2^{14} (corrects for possible channel 4 overflow since RADTIME loaded, and gives "true" time difference)

$TS_4 = K_{90mscalr}$ + RADDEL − TS_3

If $TS_4 \leq 0$:

Return

If $TS_4 + K_{mdtscalr} > 0$:

Return

Proceed to third line of "C13STALL" (a NOOP order used to avoid TC Trap)

P23

Set bit 7(RNDVZFLG) of FLAGWRD0 = 0

Set bit 1(RENDWFLG) of FLAGWRD5 = 0

If bit 13(REFSMFLG) of FLAGWRD3 = 0:

Perform "R57"

Perform "R53"

Proceed to "P23.60"

Set bit 10(SAVECFLG) of FLAGWRD9 = 0 (Tag here "P23.05"; bit used in "V94ENTER", though for clarity not shown in logic)

Perform "R57"

TS = 0570_{vn}

Perform "P23N7071"

Proceed to "V94ENTER"

<u>P23N7071</u> (Entered from "P23" for N70 and "P23.60" for N71)

22SUBSCL = TS (0570_{vn} or 0571_{vn})

TS = 22SUBSCL

Proceed to "VNFLASH": (if terminate, proceed to "GOTOPOOH")
if proceed, proceed
otherwise, proceed to previous line

Set bit 12(LUNLATLO) of FLAGWRD3 = 1

If LANDMARK ≠ 0:

If HORIZON ≠ 0, proceed to "R23.10"

TS = LANDMARK

If LANDMARK = 0:

If HORIZON = 0, proceed to "R23.10"

TS = HORIZON

If bits 9-7 (digit "C") of TS ≠ 1 or ≠ 2:

Proceed to "R23.10"

If bit 7 of TS = 1: (digit C = 1)

Set bit 12(LUNLATLO) of FLAGWRD3 = 0

Set bit 11(FARHOR) of FLAGWRDO = 1 (Tag here "P23.176")

If HORIZON = 0:

Perform "P22SUBRB"

If HORIZON ≠ 0:

If bits 6-4 (digit "D") of HORIZON ≠ 1 or ≠ 2:

Proceed to "R23.10"

If bit 4 of HORIZON = 1: (digit D = 1)

Set bit 11(FARHOR) of FLAGWRDO = 0

If STARCODE = $\pm$ 0: (Tag here "P23.30"; N88 if 77777_8 too)

TS = 0688_{vn} (Tag here "LDPLANET")

Proceed to "VNFLASH": (if terminate, proceed to "GOTOPOOH")
if proceed, proceed
otherwise, proceed to previous line

$\underline{STARSAV2} = unit(K_{1dsqr3}\ \underline{STARSAV3})$

Return

If STARCODE $\leq$ 0: (only effective if $<$ 0)

Proceed to "R23.10"

If (STARCODE - 45_8) $>$ 0:

Proceed to "R23.10"

BESTI = 6 STARCODE (Tag here "P23.170")

$\underline{STARSAV2} = \underline{K}_{cat_{BESTI}}$

Return

R23.10

Set bit 7(Operator error) of channel 11 = 1

Proceed to second line of "P23N7071"

V94ENTER (Entered from "P23" and due to restart group 2.11, set by "VERB94": tag for machine-language entrance is "V94NTR")

MARKTIME = T_{now}

Perform "POINTAXS"

$\underline{UCLSTAR} = \underline{RCLL}$

TS = 00202_8

Proceed to "GOPERF1": if terminate, proceed to "GOTOPOOH"
if proceed, proceed
otherwise, proceed to "DOVECPT"

$\underline{Z}_{sm} = unit\underline{UCLSTAR}$

$\underline{UCLSTAR} = unit(\ unit\underline{Z}_{sm} + K_{onedc}\ VZ\underline{C})$

$\underline{TS} = unit\left(\underline{STARSAV2} + K_{onedc}(VZ\underline{C} - \underline{C}_{veso})\right)$ (continue onward if SAVECFLG = 0, as it should)

$\underline{Y}_{sm} = \text{unit}(\underline{TS} * \underline{Z}_{sm})$ (Tag here "OCCLCOMP")

$\underline{X}_{sm} = \text{unit}(\underline{Y}_{sm} * \underline{Z}_{sm})$

$\underline{X}_{dc} = \text{unit}(\text{ NB1NB2}_0 \; [X_{sm}])$

$\underline{Y}_{dc} = \text{unit}(\text{ NB1NB2}_3 \; [X_{sm}])$

$\underline{Z}_{dc} = \text{unit}(\text{ NB1NB2}_6 \; [X_{sm}])$

$[X_{sm}] = [\text{REFSMMAT}]$

Perform "CALCGA"

Set bit 6(3AXISFLG) of FLAGWRD5 = 1

Perform "DOR60" (starting at 4th line)

Proceed to "P23.57" (after updating restart group 2)

DOVECPT

$\text{T}\underline{\text{S}}$ = UCLSTA$\underline{\text{R}}$

Perform "DOR60"

Proceed to "P23.57" (after updating restart group 2)

DOR60

$\text{POINTVS}\underline{\text{M}} = \text{unit}([\text{REFSMMAT}] \; \text{T}\underline{\text{S}})$

$\text{SCAXI}\underline{\text{S}} = \underline{K}_{jcaxis}$

Set bit 6(3AXISFLG) of FLAGWRD5 = 0

SRTEMP = Return address (Tag here "DOR60B")

CADRFLSH+2 = "R60CALL" + 3

Set restart group 2 = phase 1 (i.e. 2.1, causing "R60CALL" to be entered via "INITDSP" logic if a restart)

Proceed to "R60CALL"

R60CALL

Perform "R60CSM"

Proceed to address specified by SRTEMP

P23.57

Set bit 9(LMKTRG) of FLAGWRD1 = 0

Set bit 11(V94FLAG) of FLAGWRD9 = 1

Perform "R52" (uses STARSAV$\underline{2}$ since STARIND set 0 in "R00")

Set bit 11(V94FLAG) of FLAGWRD9 = 0

Proceed to "P23.60"

P23.60 (Entered from "P23" if REFSMFLG = 0, and from "P23.57")

Inhibit interrupts

MARKTIME = MRKBUF1_{dp}

TRUNION = MRKBUF1+5

Release interrupts

Inhibit interrupts

MARKDOWN+i = MRKBUF1+i (i = 0 - 6)

Release interrupts

Set bit 10(SAVECFLG) of FLAGWRD9 = 1 (used in "P23.85", though for clarity not shown in logic)

TS = 0571_{vn}

Perform "P23N7071"

Proceed to "P23.85"

P23.85

Perform "POINTAXS"

$\text{UCLSTA}\underline{\text{R}} = \text{unit}(\text{ unitRCL}\underline{\text{L}} + K_{onedc}\ \text{VZ}\underline{\text{C}})$ (Tag here "R23.55")

$\text{USSTA}\underline{\text{R}} = \text{unit}\left(\text{STARSAV}\underline{2} + K_{onedc}(\text{VZ}\underline{\text{C}} - \underline{\text{C}}_{veso})\right)$ (continue onward if SAVECFLG = 1, as it should)

$\text{COSQ} = \text{USSTA}\underline{\text{R}} \cdot \text{UCLSTA}\underline{\text{R}}$

$\text{BVECTO}\underline{\text{R}}_0 = \text{unit}(\text{USSTA}\underline{\text{R}} - \text{COSQ UCLSTA}\underline{\text{R}})$

$\text{BVECTO}\underline{\text{R}}_1 = 0$

$\text{BVECTO}\underline{\text{R}}_2 = 0$

TS_1 = TRUNION, converted to ones complement form

TS_2 = TS_1 - TRUNBIAS, rescaled to scale factor B0 revolutions

$TS = K_{2pi} \left|\underline{RCLL}\right| (TS_2 - \cos^{-1} COSQ + K_{trun19})$ (scaled B29)

If bit 12(CMOONFLG) of FLAGWRD8 = 1:

    Shift TS left 2 places (to B27 scaling)

DELTAQ = TS (Tag here "R23.51")

$VARIANCE = \left|\underline{RCLL}\right|^2 (K_{trunvar}) + K_{varsubl}$

Set bit 9(DMENFLG) of FLAGWRD5 = 0 (means 6-dimensional incorporation)

Perform "INCORP1"

$TS = \left|\underline{DELTAX}_1\right|$

If bit 12(CMOONFLG) of FLAGWRD8 = 1:

    Shift TS right 2 places (to B7)

N49DISP+2 = TS

$TS = \left|\underline{DELTAX}_0\right|$

If bit 12(CMOONFLG) of FLAGWRD8 = 1:

    Shift TS right 2 places (to B29)

N49DISP = TS

$TS = 0649_{vn}$

Perform "GOFLASHR": if terminate, proceed to "GOTOPOOH"
if proceed, skip next 2 lines
otherwise, proceed to "GOTOPOOH"

$TS = 100_2$ and perform "BLANKET" (R3BLNK)

End of job

Set bit 8(CSMUPDAT) of FLAGWRD1 = 1

Perform "INCORP2" (no time tag change from original integration done by "POINTAXS" at start of "P23.85")

Proceed to "GOTOPOOH"

POINTAXS

If bit 6(ORBWFLAG) of FLAGWRD3 = 0:

$W_i = 0$ (i = 0 - 17) (i.e. $[W_0]$ and $[W_1]$)

$W_i = 0$ (i = 27 - 44) (i.e. $[W_3]$ and $[W_4]$)

W_i = WMIDPOS (i = 0, 4, 8, diagonal elements of $[W_0]$)

W_i = WMIDVEL (i = 36, 40, 44, diagonal elements of $[W_4]$)

Perform "SETINTG"

If bit 6(ORBWFLAG) of FLAGWRD3 = 1:

Set bit 1(WMATINT) of FLAGWRD3 = 1

Set bit 6(ORBWFLAG) of FLAGWRD3 = 1

Perform "INTEGRV"

If bit 12(CMOONFLG) of FLAGWRD8 = 1: (Tag here "RECT.1")

X2 = -2

If bit 12(LUNLATLO) of FLAGWRD3 = 1:

Set bit 10(ZMEASURE) of FLAGWRDO = 0 (state and measurement bodies same)

If bit 12(LUNLATLO) of FLAGWRD3 = 0:

Set bit 10(ZMEASURE) of FLAGWRDO = 1

If bit 12(CMOONFLG) of FLAGWRD8 = 0:

X2 = 0 (Tag here "RECT.3")

If bit 12(LUNLATLO) of FLAGWRD3 = 1:

Set bit 10(ZMEASURE) of FLAGWRDO = 1

If bit 12(LUNLATLO) of FLAGWRD3 = 0:

Set bit 10(ZMEASURE) of FLAGWRDO = 0

$T\underline{S} = RC\underline{V}_{cm}$ + ($DELTA\underline{V}_{cm}$ shifted right 7 - X2 places) (Tag here "RECT.5")

$RZ\underline{C} = T\underline{S}$, shifted right -X2 places (to B29)

$T\underline{S} = VC\underline{V}_{cm}$ + ($NU\underline{V}_{cm}$ shifted right 4 - X2 places)

$VZ\underline{C} = T\underline{S}$, shifted right -X2 places (to B7)

If bit 10(ZMEASURE) of FLAGWRDO = 1:

TS = MARKTIME

Perform "LUNPOS"

If bit 12(CMOONFLG) of FLAGWRD8 = 0:

$T\underline{S} = - T\underline{S}$

$RZ\underline{C} = T\underline{S} + RZ\underline{C}$

If LANDMARK $\neq$ 0: (Tag here "R23.3")

Set bit 13(ERADCOMP) of FLAGWRD1 = 1

TS = MARKTIME

Perform "LALOTORV"

$T\underline{S} = ALPHA\underline{V}$

$RCL\underline{L} = T\underline{S} - RZ\underline{C}$ (Tag here "R23.5")

Return (to routine calling "POINTAXS")

Proceed to "HORIZ"

<u>HORIZ</u> (Program tag is "R23.4", but "HORIZ" retained for consistency with previous programs and mnemonic usefulness)

$T\underline{S} = (C_{unitw_x}, C_{unitw_y}, 1 - 2^{-28})$

$UBA\underline{R}_2 = unit(STARSAV\underline{2} * RZ\underline{C})$

$UBA\underline{R}_0 = unit(T\underline{S} * UBA\underline{R}_2)$

$UBA\underline{R}_1 = unit(UBA\underline{R}_2 * UBA\underline{R}_0)$

If bit 12(LUNLATLO) of FLAGWRD3 = 1:

$PDBH = K_{radmoon}$

$PDAH = K_{radmoon}$

If bit 12(LUNLATLO) of FLAGWRD3 = 0:

$ALPHAV_z = \underline{UBAR}_1 \cdot T\underline{S}$

$RATETEMP = C_{horizalt} - C_{horislp} \left| RZ\underline{C} \right|$ (RATETEMP same cells as (VHFCNT, TRKMKCNT))

Perform "GETERAD"

$PDBH = ERADM + RATETEMP$

$PDAH = K_{aearth} + RATETEMP$

$$PDR\underline{H} = \begin{bmatrix} \underline{UBAR}_0 \\ \underline{UBAR}_1 \\ \underline{UBAR}_2 \end{bmatrix} RZ\underline{C}$$ (Tag here "HORIZ.1")

$$PDUS\underline{H} = \begin{bmatrix} \underline{UBAR}_0 \\ \underline{UBAR}_1 \\ \underline{UBAR}_2 \end{bmatrix} STARSAV\underline{2}$$

$PDA = (PDRH_x / PDAH)^2 + (PDRH_y / PDBH)^2$ (divisions quasi-floating point)

$PDALPHA = (PDAH / PDBH) (\sqrt{PDA - 1} / PDA)\ PDRH_y$ (divisions each done quasi-floating point)

$PDBETA = (PDBH / PDAH) (\sqrt{PDA - 1} / PDA)\ PDRH_x$

$PDVC\underline{T}_0 = (PDRH_x / PDA + PDALPHA,\ PDRH_y / PDA - PDBETA,\ 0)$

$PDVC\underline{T}_1 = (PDRH_x / PDA - PDALPHA,\ PDRH_y / PDA + PDBETA,\ 0)$

$PDAZERO = \text{unit}(PDVC\underline{T}_0 - PDR\underline{H}) \cdot PDUS\underline{H}$

$PDAONE = \text{unit}(PDVC\underline{T}_1 - PDR\underline{H}) \cdot PDUS\underline{H}$

If PDAZERO - PDAONE < 0:

If bit 11(FARHOR) of FLAGWRD0 = 1: (Tag here "HORIZ.3")

$T\underline{S} = PDVC\underline{T}_0$

If bit 11(FARHOR) of FLAGWRD0 = 0:

$T\underline{S} = PDVC\underline{T}_1$

If PDAZERO - PDAONE $\geqslant$ 0:

If bit 11(FARHOR) of FLAGWRDO = 1:

$\underline{TS} = \underline{PDVCT}_1$

If bit 11(FARHOR) of FLAGWRDO = 0:

$\underline{TS} = \underline{PDVCT}_0$

$$\underline{TS} = \underline{TS} \begin{bmatrix} \underline{UBAR}_0 \\ \underline{UBAR}_1 \\ \underline{UBAR}_2 \end{bmatrix}$$

$\underline{RCLL} = \underline{TS} - \underline{RZC}$ (Tag here "R23.5")

Return (to routine calling "POINTAXS")

Quantities in Computations

See also list of major variables and list of routines

22SUBSCL: Single precision cell used in "P23N7071" to retain the verb-noun pattern used. Same cell also used in P22 (see Orbital and Rendezvous Navigation).

9DWI: Cell used single precision, scale factor B13, to control performance of routine to convert W matrix from 6x9 to 6x6. It is stored in push-down list location 10D.

9DWJ: Cell used single precision, scale factor B13, to control performance of routine to convert W matrix from 6x9 to 6x6. It is stored in push-down list location 14D.

9DWN: Cell used single precision, scale factor B13, to control performance of routine to convert W matrix from 6x9 to 6x6. It is stored in push-down list location 12D.

9DWP: Cell used single precision, scale factor B13, to control performance of routine to convert W matrix from 6x9 to 6x6. It is stored in push-down list location 8D.

9DWXX: Single precision cell used to retain return address information from "9DWTO6DW".

AGEOFW: Time of last W matrix initialization for rendezvous, scale factor B28, units centi-seconds. Set equal to MARKTIME in "REND5C", and can be displayed by noun 31. Used in "AUTOW" logic.

ALPHA$\underline{V}$: See Coordinate Transformations.

BESTI: See Inflight Alignment.

BVECTO$\underline{R}_0$, BVECTO$\underline{R}_1$, BVECTO$\underline{R}_2$: Vectors used as communication cells with "INCORP1" to provide geometry information associated with DELTAQ, i.e. relating the deviation in the measurement (a scalar) to the vector deviation in the state vector. The three vectors can be considered associated with position, velocity, and landmark location, and in the CSM program BVECTO$\underline{R}_1$ is always set 0. Scale factors of BVECTO$\underline{R}_0$ and BVECTO$\underline{R}_2$ (non-zero only in P22) are B1.

C_{altvar}: Single precision erasable memory constant, program notation "ALTVAR", scale factor B-16, units rad^2, used in "REND7" (part of R22) if optics backup (R23) data is used, to specify information on the a priori accuracy of the backup optics data.

$C_{horislp}$: Erasable memory (double precision) constant, program notation "HORISLP", scale factor B0, used in "HORIZ" in the determination of "horizon altitude" (it is multiplied by $|RZ\underline{C}|$ and then subtracted from the other terms).

$C_{horizalt}$: Erasable memory (double precision) constant, program notation "HORIZALT", scale factor B29, units meters, used in "HORIZ" (part of P23) to determine the "horizon altitude".

C_{intvar}: Single precision erasable memory constant, program notation "INTVAR", scale factor B15, units meters^2, added to the variance computed in "REND7" (part of R22) for "integration variance", i.e. "square of expected integration position extrapolation error".

C_{rmax}: Single precision erasable memory constant, program notation "RMAX", scale factor B19, units meters, giving the maximum value of position state vector change (as expressed in N49DISP) that is incorporated automatically by R22. Values larger than this constant cause a display for crew approval. If it is desired to generate a display for all measurement information, this can be accomplished by setting the constant to 77776_8 (one bit negative).

C_{rvar}: Erasable memory (double precision) constant, program notation "RVAR", scale factor B-16, giving the "percentage error" (squared) in the VHF ranging measurement: for VHF marks, $(|\underline{RCLP}|)^2 C_{rvar} + C_{intvar}$ is the value of VARIANCE (subject to $C_{rvarmin}$ constraint). Actually would involve "percent"/100, of course.

$C_{rvarmin}$: Triple precision erasable memory constant, program notation "RVARMIN", scale factor B40, units meters^2, giving the negative of the minimum value of VARIANCE for the VHF range measurements in R22 (negative is for coding convenience). If the most significant one-third of the constant is zero, then the least significant two-thirds can be considered as a double precision number, scale factor B26. If set very large, effectively inhibits use of VHF data(for W small).

$\underline{C}_{unitw}$: See General Program Control.

$\underline{C}_{veso}$: See Coordinate Transformations.

C_{vmax}: Single precision erasable memory constant, program notation "VMAX", scale factor B7, units meters/centi-second, giving the maximum value of velocity state vector change (as expressed in N49DISP+2) that is incorporated automatically by R22. Values larger than this constant cause a display for crew approval.

CADRFLSH+2: See Display Interface Routines.

COSQ: Cosine of angle between USSTAR and UCLSTAR, scale factor B1, stored in push-down list location 0D.

COUNT3MK: See Minimum Key Rendezvous.

CSMPOS: Value of CSM position vector, scale factor B29(earth) or B27 (moon), units meters.

DELQDA: Value of DELTAQ/TRIPA, variable scale factor, stored in push-down list location OD. If numerator and denominator are already normalized, scale factor is B-10 for earth or B-12 for moon.

DELTAQ: Value of deviation in measurement quantity, scale factor B29(earth) or B27(moon), units meters.

DELTA$\underline{V}_{cm}$, DELTA$\underline{V}_{lm}$: See Orbital Integration. Component overflows if exceeds 2^{22} meters (earth) or 2^{18} meters (moon).

DELTA$\underline{X}_0$, DELTA$\underline{X}_1$, DELTA$\underline{X}_2$: State vector deviation estimates, scale factor (for DELTA$\underline{X}_0$ and DELTA$\underline{X}_2$) B29(earth) or B27(moon), units meters; for DELTA$\underline{X}_1$, scale factor B7(earth) or B5(moon), units meters/centi-second. They are for position of vehicle, velocity of vehicle, and position of landmark respectively (i = 0,1,2).

EGRESS: Single precision cell used for storage of return address information from "INCORP2" routine.

EMATRIX$_i$: Temporary storage cells used to store intermediate results from the conversion of W matrix from 6x9 to 6x6. The cells used in the memory for this function correspond to W_{60} - W_{80}, but separate notation is used (as in the program) for clarity.

ERADM: See Coordinate Transformations.

GAMMA: Quantity computed in "INCORP1", variable scale factor (scale factor information partially contained in cell NORMGAM). If NORMGAM = 0, scale factor of GAMMA equivalent to B-40. NORMGAM subsequently modified to reflect rescaling of $\underline{Z}_i$ (for use in "INCORP2").

HORIZON: Single precision quantity loaded and displayed in R3 of N70 and N71, used only in P23 (see "P23N7071"). If LANDMARK is non-zero, then HORIZON must be zero; if LANDMARK = 0, then HORIZON checked to be of the form XXCDX, where C = 1 means earth and C = 2 means lunar (other values not allowed); D = 1 means near horizon and D = 2 means far horizon (other values not allowed). If an illegal input is detected (the X's may be of any value, provided LANDMARK = 0), then the display is generated again, after energizing the operator error light.

K_{1dsqr3}: Constant, program notation "1DSQR3", scale factor B0, value 0.57735021. Value corresponds approximately to (1/1.732051), or the reciprocal of $\sqrt{3}$.

K_{2pi}: Constant, program notation "PI/4.0", scale factor B3, value 0.785398164. Value corresponds to 2 x 3.141592656 x 2^{-3}, or approximately 2π x 2^{-3}, to convert between angle measurements in units of revolutions and radians.

$K_{60secdp}$: Constant, program notation "60SECDP", scale factor B28, units centi-seconds. Value is 6000 x 2^{-28}, corresponding to 60 seconds: if T_{now} - VHFTIME is at least this amount, then an attempt can be made to obtain a new VHF mark in "REND3".

$K_{90mscalr}$: Single precision constant, program notation "90MSCALR", scale factor B9, units centi-seconds. Value is 440_8, or 9×2^{-9}, corresponding to 9 centi-seconds (the time delay, when added to RADDEL, between initiation of radar read and completion of the shift pulses).

K_{aearth}: Constant, program notation "AEARTH", scale factor B29, units meters. Value is 6378166×2^{-29}, the Fischer model equatorial earth radius.

$\underline{K}_{cat}$: See Inflight Alignment.

$K_{convrnge}$: Constant, program notation "CONVRNGE", scale factor B13, units of (meters/bit). Value is 18.52×2^{-13}, corresponding to a VHF ranging system bit weight of 18.52 meters (or 0.0100 nmi). A value of (2^{14} $K_{convrnge}$) corresponds to the weight of bit 15, normally the "sign", of RM.

K_{ft999}: Constant, program notation "FT99999", scale factor B19, units meters. Value is 30479×2^{-19}, corresponding to $99996.72 \times 0.3048 \times 2^{-19}$, where first term is value in feet (display would be expected to show 99997 feet), second converts to meters, and third is scale factor.

K_{imuvar}: Constant, program notation "IMUVAR", scale factor B-16, units radians^2. Value is 0.04E-6 x 2^{16}, corresponding to 0.04 mr^2.

$\underline{K}_{jcaxis}$: Vector constant, program notation "JCAXIS", scale factor B1, value (0.268819062, 0, 0.421587846). True values correspond to (0.537638124, 0, 0.843175692), or (sin θ, 0, cos θ), where the angle is approximately the same as that defined in [NB1NB2] (see Coordinate Transformations): the octal matrix element for the sine is one least increment larger than that in $\underline{K}_{jcaxis}$.

$K_{mdtscalr}$: Single precision constant, program notation "-DTSCALR", scale factor B9, units centi-seconds. Value is 77754_8, corresponding to -23_8, or -0.59375 centi-seconds (a measure of when the shift pulses are sent to the radar).

K_{onedc}: Constant, program notation "ONE/C", scale factor B-21, units of (meters/centi-second)$^{-1}$. Value is 0.333564049E-6 x 2^{21}, corresponding to (1/2.997925E6) x 2^{21}, where first term is reciprocal of speed of light in meters/centi-second (corresponds to 9.835712E8 fps) and second is scale factor.

$K_{radmoon}$: Constant, program notation "RADMOON", scale factor B29, units meters. Value is 1738090×2^{-29}, the mean lunar radius.

K_{sxtvar}: Constant, program notation "SXTVAR", scale factor B-16, units radians^2. Value is 0.04E-6 x 2^{16}, corresponding to 0.04 mr^2.

K_{trun19}: Constant, program notation "TRUN19", scale factor B0, units revolutions. Octal value is 01604_8 00000_8, used to correct TRUNION for the built-in bias. Decimal value is 900×2^{-14}, or $(7200 \times 2^{-3}) \times 2^{-14}$, where the 7200 is the complement of the optics trunnion "zero" value (see K_{20degs} in Optics Computations), and the 2^{-3} is scale factor compensation (from B-3 to B0 revolutions).

$K_{trunvar}$: Constant, program notation "TRUNVAR", scale factor B-18, units radians^2. Value is 2.5E-9 x 2^{18}, corresponding to 0.0025 mr^2.

$K_{varsubl}$: Triple precision constant, program notation "VARSUBL", scale factor B40, units meters^2. Value is 3.429904E6 x 2^{-40}, corresponding to $(1852)^2 \times 2^{-40}$, or the square of one nautical mile expressed in meters. In the listing, program sets the most significant one-third of the constant 0 (and uses the fact that it is zero elsewhere), and treats the least significant two-thirds as a double precision number, scale factor B26 (see $C_{rvarmin}$).

LANDMARK: See Orbital and Rendezvous Navigation.

LEMPOS: Value of LM position vector, scale factor B29 (earth) or B27 (moon), units meters.

MARKDOWN+i (i = 0-6): Set of optics mark information (double precision time, CDU_y, optics shaft, CDU_z, optics trunnion, and CDU_x respectively) intended for downlink transmission in programs making use of optics data. For P23, cells loaded in "P23.60" with measurement data, and in "MARKDIF" if get an R57 mark. Address is selected in "SXTSM" if STARIND = 0. For P51 - P54, this means the BESTI body being marked; for P22 and R22, since STARIND set 0 in "R00", the cells are also loaded (due to performance of part of "SXTSM" via "GETUM"). MARKDOWN+7 is the same cell as RM.

MARKTIME: Cells used to contain time of mark information for R22 and P23, and as a communication cell with "SETINTG" (for which only first 2 cells are employed, scale factor B28, units centi-seconds). Used in R22 as the first 2 cells of a 7-cell buffer to contain optics mark data being processed, as loaded from MRKBUF1 cells in "REND1".

MRKBUF1: See Optics Computations.

N49DISP: Magnitude of position state vector change for display in R1 of N49, scale factor B29, units meters. The magnitude is calculated as the square root of the sum of the squares of individual components of $\underline{DELTAX}_0$, where the squares are carried triple precision. Hence if a given component is of insufficient magnitude to be retained triple precision when squared, that component will contribute 0 to the sum: since for triple precision scaled B0 the least increment is 2^{-42}, this means that for a component to have an input it must be at least 2^{-21}. For earth scaling, this would correspond to 2^8 (256) meters and for moon scaling 2^6 (64) meters. Hence if all three components of $\underline{DELTAX}_0$ are less than 256/64 meters (earth/moon), N49DISP will display zero; 256 meters is about 0.14 nmi, while R1 of N49 least increment is 0.01 nmi.

N49DISP+2: Magnitude of velocity state vector change for display in R2 of N49, scale factor B7, units meters/centi-second. Magnitude is calculated in a manner analogous to N49DISP, using $\underline{DELTAX}_1$ instead of $\underline{DELTAX}_0$. The effect, however, is considerably less, since for earth scaling least increment on square is 2^{-14} meters/centi-second and for moon 2^{-16} meters/centi-second: these correspond to about 0.02 fps and about 0.005 fps respectively, while the display least increment is 0.1 fps. See C_{vmax}.

N49DISP+4: Single precision cell, scale factor B14, used to display in R3 of N49 (for R22 only) information on the source of the update generating the display. It is set to 1 in "REND7" for optics data and 2 for VHF ranging data.

[NB1NB2]: See Coordinate Transformations. First row is $\underline{NB1NB2}_0$, second row is $\underline{NB1NB2}_3$, and third row is $\underline{NB1NB2}_6$.

$NU\underline{V}_{cm}$: See Orbital Integration.

OLDMKTME: Time of previous mark, scale factor B28, units centi-seconds, loaded in "REND12" and used in "AUTOW" logic.

$\underline{OMEGA}_0$, $\underline{OMEGA}_1$, $\underline{OMEGA}_2$: Weighting vectors for measurement incorporation of DELTAQ measurement (to give a "statistically optimum linear estimate of the deviation from the estimated state vector"). Scale factors of $\underline{OMEGA}_0$ and $\underline{OMEGA}_2$ are B39, while the scale factor of $\underline{OMEGA}_1$ is B20. This scaling is for no $\underline{Z}_i$ rescaling.

$\underline{OMEGAM}_0$, $\underline{OMEGAM}_1$, $\underline{OMEGAM}_2$: Values of $\underline{OMEGA}_i$ multiplied by GAMMA, used in "INCORP2" for updating W matrix information. Scale factor depends on scaling of GAMMA: if NORMGAM = 0 (see GAMMA), then scale factor of $\underline{OMEGAM}_0$ and $\underline{OMEGAM}_2$ is B-1, and for $\underline{OMEGAM}_1$ scale factor is B-20. Program notations for these quantities are "OMEGAM1", "OMEGAM2", and "OMEGAM3" respectively.

OPTMODES: See Optics Computations.

PBODY: See Orbital Integration.

PDA: Value of intermediate quantity "A" in "HORIZ", scale factor B18, stored in push-down list location 16D. The ratio $(\sqrt{A - 1})/A$ is stored in push-down list location 28D, scale factor B0.

PDAH: Value of semi-major axis used to define shape of horizon ellipse in "HORIZ", scale factor B29, units meters, stored in push-down list location 2D.

PDALPHA: Value of intermediate quantity computed in "HORIZ", scale factor B29, units meters, stored in push-down list location 24D.

PDAONE: Value of intermediate quantity "A_1" computed in "HORIZ", scale factor B2, stored in push-down list location 24D.

PDAZERO: Value of intermediate quantity "A_0" computed in "HORIZ", scale factor B2, stored in push-down list location 22D.

PDBETA: Value of intermediate quantity computed in "HORIZ", scale factor B29, units meters, stored in push-down list location 26D.

PDBH: Value of semi-minor axis used to define shape of horizon ellipse in "HORIZ", scale factor B29, units meters, stored in push-down list location 0D.

PDR$\underline{H}$: Value of RZ$\underline{C}$ transformed to Horizon Coordinate System, scale factor B29, units meters, stored in push-down list location 4D.

PDUS$\underline{H}$: Value of STARSAV$\underline{2}$ transformed to Horizon Coordinate System, scale factor B1, stored in push-down list location 10D.

PDVC$\underline{T}_0$: Vector defining "point of tangency from PDR$\underline{H}$ to horizon", scale factor B29, units meters, stored in push-down list location 28D.

PDVC$\underline{T}_1$: Vector defining the other point of tangency to horizon, scale factor B29, units meters, stored in push-down list location 16D.

POINTEX: Single precision cell used to retain return address information from "WAITONE".

POINTVS$\underline{M}$: See Attitude Maneuvers.

QPRET: See Orbital Integration.

QREG: Computer Q register (cell 0002_8), loaded with S-register information for computer TC orders, and also used in some cases for temporary storage of other information. As used in "RENDISP2", apparently expected to contain a positive non-zero number (as would be true for a TC order use). The actual sampling of QREG takes place in the program service routine "BANKJUMP", which is entered at the end of "ENDRET2"to return to the "PRIODSP" caller.

$\underline{R}_{other}$: See Orbital Integration.

R61CNTR: See Orbital and Rendezvous Navigation.

RADDEL: Single precision time until the 100 pps pulse initiating radar read sequence (loaded in "RADSTART"), scale factor B9, units centi-seconds.

RADTIME: Single precision storage for the complement of the contents of channel 4, scale factor B9, units centi-seconds, as sampled when bit 4 of channel 13 was set to initiate radar reading.

RATETEMP: Temporary storage cell used in "HORIZ" to contain horizon-altitude correction information, scale factor B29, units meters. TRKMKCNT is the least significant half of the cell and VHFCNT is the most significant half. Also see Optics Computations.

RCL$\underline{L}$: Tag assigned to value of (R$\underline{L}$ - RZ$\underline{C}$) in this document, scale factor B29, units meters. Left in MPAC when exit from "POINTAXS"/"HORIZ".

RCL$\underline{P}$: Value of (LEMPO$\underline{S}$ - CSMPO$\underline{S}$) for R22, scale factor B29 (earth) or B27 (moon), units meters.

RC$\underline{V}$, RC$\underline{V}_{cm}$, RC$\underline{V}_{lm}$: See Orbital Integration.

RM: Single precision value of measured range sampled from RNRAD cell, scale factor B14, units counts (one VHF count is 0.01 nmi). The cell is the same as MARKDOWN+7 (for telemetry purposes).

RNRAD: Single precision special erasable memory cell 0046_8, into which VHF range data is shifted under hardware control if bits 4-1 of channel 13 are set to 1001_2. When shifting is complete, program interrupt #9 is generated, causing entrance to "VHFREAD".

RZ$\underline{C}$: Value of position vector to the body with which the horizon or landmark measurement in P23 is made, scale factor B29, units meters.

SCAXI$\underline{S}$: See Attitude Maneuvers.

SRTEMP: Exit address (single precision) from "R60CALL".

STARCODE: See Inflight Alignment. Mentioned in comments of this section.

STARIND: See Inflight Alignment.

STARSAV$\underline{2}$: See Inflight Alignment: in P23 used for $\underline{K}_{cat}$ (or N88) information which is employed by R52, with notation "US".

STARSAV$\underline{3}$: See Inflight Alignment.

T_{et}, TDELTA$\underline{V}$: See Orbital Integration.

TEMPOR1: Single precision cell, scale factor B14, used in the "RENDISP"/"RENDISP2" interface to determine when an answer to the 0649 display is received (when value becomes non-zero) and to conclude that this answer was PRO if negative and that it was a recycle if it was positive (see QREG). This method is employed to avoid display conflicts (with e.g. R60 P20 display), and will cause the computer activity light to remain on until the display is answered.

TNU$\underline{V}$: See Orbital Integration.

TRIPA: Triple precision value of sum of VARIANCE and $|\underline{Z}_i|^2$ computed in "INCORP1", scale factor B40. It is used as the denominator value for computing the necessary weighting functions (a rounded version is used to multiply VARIANCE).

TRKMKCNT: Single precision cell, scale factor B14, giving a count of the number of optics marks (either primary optics or via R23) that have been completely incorporated into the state vector in R22 (it is incremented in "REND12"). It is set 0 in "DOFSTART", "P37E", and in "REND5C" (when initialize W matrix for R22 use). Cell is the "least significant half" of VHFCNT, and can be displayed as the 2 least significant digits in R3 of N45. The same cell is also used for RATETEMP+1 (see this writeup and Optics Computations).

TRUNBIAS: See Optics Computations.

TRUNION: Single precision value of measured optics trunnion angle for P23, loaded in "P23.60", scale factor B-3, units revolutions.

TX78$\underline{9}$: Temporary storage in "INCORP2" for updated value of X78$\underline{9}$, same scaling (not loaded into X78$\underline{9}$ unless bit 9(DMENFLG) of FLAGWRD5 = 1, as it would be for P22 but not for P23 or R22 of P20).

$UBA\underline{R}_0$, $UBA\underline{R}_1$, $UBA\underline{R}_2$: Unit vectors, scale factor B1, used in "HORIZ" to define the transformation matrix from reference coordinates to the Horizon Coordinate System.

UC$\underline{L}$: Value of unitRCL$\underline{P}$ computed in "REND7", scale factor B1 (a unit vector, of course).

UCLSTA$\underline{R}$: Unit vector information, scale factor B1, used in "P23.85" and "V94ENTER" for corrected vector to landmark/horizon. Cell also used for temporary storage purposes.

U$\underline{M}$: Value of unit measurement vector derived in "GETUM", scale factor B1: it gives the direction of the optics-measured information in reference coordinates.

USSTA$\underline{R}$: Value of unit vector to celestial body computed in "P23.85", scale factor B1 (includes correction for velocity of light).

USTA$\underline{R}$: Unit "fictitious star direction", scale factor B1, updated in "BVECTORS" (initialized in "REND7" or P22). Two such quantities are used, "chosen to be perpendicular to each other and to the current estimated line-of-sight vector."

$\underline{V}_{other}$: See Orbital Integration.

VARIANCE: Triple precision variance associated with the navigation measurement reflected in DELTAQ, scale factor B40, units of meters^2. The value for use in "INCORP1" is computed by the individual calling program based on a priori measurement errors.

VC$\underline{V}$, $VC\underline{V}_{cm}$: See Orbital Integration.

VHFCNT: Single precision cell, scale factor B14, giving a count of the number of VHF range marks that have been incorporated into the state vector in R22 (it is incremented in "REND12"). It is reset to 0 at the points listed for TRKMKCNT (which is the "least significant half" of VHFCNT), and it can be displayed as the 2 most significant digits in R3 of N45. Same cell used for most significant half of RATETEMP (see this writeup and Optics Computations).

VHFRANGE: Value of incoming VHF range measurement (a scaled version of the 15-bit integer in RM), scale factor B27 (earth or moon), units meters, computed in "RANGERD1". If reading unsuccessful, used instead to control lighting of Tracker light in "LIGHTON".

VHFTIME: Value of computer clock when last VHF range sample was made, scale factor B28, units centi-seconds. It is initialized to present computer clock in "R22", and if more than $K_{60secdp}$ have elapsed since it was last loaded (see "REND3" logic), then another sample attempt is initiated. It is set to MARKTIME in "LIGHTON" (sample not successful) or "RANGERD1" (sample successful): MARKTIME is set to T_{now} in "RADSTART" (just after setting channel 13 bits to initiate sample, but <u>after</u> releasing interrupts that were inhibited during channel 13 setting).

$\underline{VZC}$: Value of vehicle velocity vector for P23 computed in "POINTAXS", scale factor B7, units meters/centi-second (same scaling for earth or moon).

$[W_i]$, W_i , $\underline{W}_i$: Notations used to describe portions of the "W" (error transition) matrix stored in the computer memory: the complete matrix consists of 9 rows and 9 columns, giving 81 elements. Each element in the memory is stored double precision, so that the last element would have the notation "W +160" in the program. For simplicity, however, this incrementing of addresses by 2 has been suppressed in the programmed equations (indices are shown scaled B13 instead of B14), so that the first element is called W_0, the next W_1, and the last element W_{80}. Computations carried out by vector operations on three elements at a time are designated by the usual vector notation with the subscript being that of the first element involved: $\underline{W}_3$, for example, is the "vector" made up of (W_3 , W_4 , W_5).

Other manipulations in the program make use of the 3x3 matrix computation capabilities of the interpretive language. For convenience in description, the indices for these 3x3 matrices run from 0 to 8 (all matrix operations shown are 3x3), with the index determined by the index of the first element divided by 9: $[W_2]$ would be the matrix with rows made up of $\underline{W}_{18}$, $\underline{W}_{21}$, and $\underline{W}_{24}$, for example. The complete 9x9 matrix is made up of these 3x3 matrices, with the first three rows being $[W_0]$, $[W_1]$, and $[W_2]$; the middle three rows $[W_3]$, $[W_4]$, and $[W_5]$; and the final three rows $[W_6]$, $[W_7]$, and $[W_8]$. For a 6x6 (P20 and P23) instead of 9x9 (P22) W matrix application, only $[W_0]$, $[W_1]$, $[W_3]$, and $[W_4]$ are employed.

Initialization of the W matrix involves initializing the diagonal elements of $[W_0]$ with the appropriate position uncertainty information and the diagonal elements of $[W_4]$ with the appropriate velocity uncertainty information: see P22 computations for the initialization performed for landmark information.

The scale factor of all elements of a given 3x3 matrix is the same. For matrix $[W_i]$, with i = 0,1,2,6,7,8, the scale factor is B19 in units of meters; for i = 3,4,5, the scale factor is B0 in units of meters/centi-second.

WMIDPOS: Single precision value of W matrix initialization for position information ($[W_0]$), scale factor B19, units meters. Can be updated by "V67CALL", and also would be expected to form part of prelaunch load: it is used in "POINTAXS" to initialize the W matrix for P23.

WMIDVEL: Single precision value of W matrix initialization for velocity information ($[W_4]$), scale factor B0, units meters/centi-second. Can be updated by "V67CALL", and also would be expected to form part of prelaunch load: it is used in "POINTAXS" to initialize the W matrix for P23.

WORBPOS: Single precision value of W matrix initialization for position information, scale factor B19, units meters. Can be updated by "V67CALL", and also would be expected to form part of prelaunch load: it is used in P22 for W matrix initialization.

WORBVEL: Single precision value of W matrix initialization for velocity information, scale factor B0, units meters/centi-second. Can be updated by "V67CALL", and also would be expected to form part of prelaunch load: it is used in P22 for W matrix initialization.

WORKW: Temporary storage cell used in the conversion of a 6x9 matrix to 6x6 form. Corresponds to push-down list location 0D.

WRENDPOS: Single precision value of W matrix initialization for position information, scale factor B19, units meters. Can be updated by "V67CALL", and would be expected to form part of the prelaunch load. It is also loaded (if final pass and in minimum key rendezvous mode) in "VN1645" with $K_{posvel2}$ (see Burn Computations). It is used in "REND5C" to initialize the W matrix for R22.

WRENDVEL: Single precision value of W matrix initialization for velocity information, scale factor B0, units meters/centi-second. Can be updated by "V67CALL", and would be expected to form part of the prelaunch load. It is also loaded (if final pass and in minimum key rendezvous mode) in "VN1645" with $K_{posvel3}$ (see Burn Computations). It is used in "REND5C" to initialize the W matrix for R22.

WWOPT: Single precision cell, scale factor B14, used in "V67CALL" to specify whether the matrix initialization elements are to be changed, and if so which set. The value when displayed initially in R3 of N99 is 0. A non-zero value means a change desired, with a value of 1 (or negative non-zero) changing WRENDi; a value of 2 changing WORBi; and a value in excess of 2 changing WMIDi.

WWPOS: Quantity displayed in R1 of N99, scale factor B19, units meters. It provides the present W matrix position error. If the loading option is selected, is used as source for WiPOS information.

WWVEL: Quantity displayed in R2 of N99, scale factor B0, units meters/centi-second. It provides the present W matrix velocity error. If the loading option is selected, is used as source for WiVEL information.

$X78\underline{9}$: Value of estimated landmark location, scale factor B29 (earth) or B27(moon), units meters, as updated by results of measurement incorporation routines.

$\underline{Z}_0$, $\underline{Z}_1$, $\underline{Z}_2$: Measurement vector information, scale factor B20, units meters, computed at the start of "INCORP1". Subsequently rescaled.